

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Disseny i desenvolupament d'un lloc web per al projecte d'un centre social

MEMÒRIA

Autora: Laura Planells Hernández
Director: Oriol Boix
Convocatòria: Juny de 2018



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Un nou col·lectiu vol crear un espai social on s'hi facin activitats i on s'hi ofereixin altres serveis. En aquest treball es desenvoluparà el lloc web amb el que es donarà a conèixer l'entitat i amb el que es podrà gestionar part de la seva feina. Es farà tenint en compte les necessitats de les persones que l'utilitzaran i se centrarà en la part estructural.

Els requisits de disseny s'han avaluat fent una anàlisi de les necessitats de les possibles persones usuàries del web, separant-les pels rols que prendran i per les característiques grupals d'aquestes persones. També s'ha tingut en compte el canvi de legislació en matèria de protecció de dades. Altres requisits són els plantejats per la pròpia entitat i els relacionats amb aspectes tècnics.

Per a portar-ho a terme, s'han seleccionat les eines per realitzar la programació; començant per definir l'entorn de treball, que condiciona la resta del disseny. S'ha definit quina estructura ha de tenir el lloc web, tant pel que fa a l'estructura de carpetes com per la de les pàgines. S'ha creat una base de dades i s'ha desenvolupat les característiques de cada taula i la relació entre elles. S'han generat els arxius referents a la lògica del lloc web, dividida principalment entre models i controladors, i els arxius per les vistes, on hi ha la part visual.

També s'han destacat les principals accions realitzades per a la seguretat del lloc web, s'ha resumit quin programari de suport s'ha utilitzat i s'ha fet un pressupost i una anàlisi de l'impacte social i ambiental.

Com a conclusions, s'ha aconseguit crear un lloc web que complís els objectius marcats, que servirà de suport a l'equip de l'entitat, ajudarà a gestionar-ne la informació i serà útil per a les persones que l'utilitzin, s'ha creat l'estructura del web, s'ha triat el programari necessari, s'ha creat una base de dades i els arxius necessaris per gestionar-la i per introduir i mostrar la informació. Tot això s'ha fet tenint en compte la legalitat vigent.

Sumari

RESUM	1
SUMARI	3
1. INTRODUCCIÓ	7
1.1. Problemàtica	7
1.2. Objectius del projecte	7
1.3. Abast del projecte	8
2. PERSONES USUÀRIES	9
2.1. Responsables de l'entitat	9
2.2. Persones usuàries de l'espai	10
2.2.1. Perfil de persona usuària habitual	10
2.2.2. Perfil de persona que assisteix a activitats puntuals	10
2.2.3. Perfil de persona que ofereix activitats	10
3. REQUISITS DE DISSENY	12
3.1. Necessitats de les persones usuàries	12
3.2. Normativa	12
3.3. Requisits de l'entitat	13
3.4. Aspectes tècnics	14
4. EINES DE PROGRAMACIÓ	15
4.1. Laravel	15
4.2. Panell d'administració	16
4.3. Bootstrap	17
4.4. Formulari	17
4.5. Sistema de registre i verificació de persones usuàries	18
5. ESTRUCTURA	19
5.1. Estructura de carpetes	19
5.2. Menú principal	21
5.3. Estructura del panell d'administració	22
5.4. Capçalera i peu de pàgina	24
6. BASE DE DADES	25
6.1. Taula de persones usuàries	26
6.2. Rols, permisos i relacions entre taules	27

6.3.	Taula de categories	28
6.4.	Taules d'activitats i de tipus de públic.....	29
6.5.	Taula d'articles.....	31
6.6.	Taula de sales i espais	32
6.7.	Taula per al directori	32
6.8.	Altres taules	34
7.	MODELS I CONTROLADORS	35
7.1.	Models	35
7.2.	Notificador.....	37
7.3.	Rutes.....	37
7.4.	Polítiques d'autorització	38
7.5.	Controladors	39
7.5.1.	Controlador per als articles	39
7.5.2.	Controlador per a les activitats.....	40
7.5.3.	Controlador per al directori.....	41
7.5.4.	Controlador per al perfil.....	42
7.5.5.	Controlador per a la pàgina principal	42
8.	VISTES	44
8.1.	Plantilles i parcials.....	44
8.2.	Error de visualització	45
8.3.	Pàgina principal	45
8.4.	Vistes per al perfil.....	45
8.5.	Vistes de registre i inici de sessió	46
8.6.	Vistes per a les activitats.....	47
8.7.	Vistes per als articles	48
8.8.	Vistes per al directori.....	48
8.9.	Vistes de les pàgines de presentació	49
8.10.	Vistes de les pàgines de serveis.....	50
8.11.	Vista per a les condicions d'ús.....	50
9.	SEGURETAT I TRACTAMENT DE DADES	51
9.1.	Normativa.....	51
9.2.	Contrasenya i encriptació	52
9.3.	Protocol (HTTPS).....	52
9.4.	Mesures de prevenció d'atacs	53

10. DISSENY DE L'ESTIL	54
10.1. Arxius css d'estil	54
10.2. Disseny visual.....	54
11. PROGRAMARI UTILITZAT	55
11.1. Editor de codi.....	55
11.2. Administrador de bases de dades.....	55
11.3. Fals servidor de correu electrònic	56
11.4. Programari de suport a <i>Laravel</i>	56
12. PRESSUPOST	57
CONCLUSIONS	59
AGRAÏMENTS	61
BIBLIOGRAFIA	62
Referències bibliogràfiques.....	62
Bibliografia complementària.....	64
ANNEX	67
A.1. Rutes	69
A.2. Models	71
A.2.1. Activitat	71
A.2.2. Directori.....	72
A.2.3. Publics	72
A.2.4. Sales.....	73
A.2.5. User	73
A.3. Vistes.....	74
A.3.1. index	74
A.3.2. activitats/activundia	76
A.3.3. activitats/curs	77
A.3.4. activitats/jornades	77
A.3.5. activitats/allactivitats.....	78
A.3.6. activitats/senseactivitats.....	79
A.3.7. activitats/crear	79
A.3.8. activitats/editar	82
A.3.9. activitats/partials/botons	85
A.3.10. activitats/partials/assistents.....	86
A.3.11. articles/allposts.....	87
A.3.12. articles/post.....	88

A.3.13. directori/index.....	88
A.3.14. directori/senseentrades.....	90
A.3.15. directori/nova-entrada	91
A.3.16. directori/editar	93
A.3.17. perfil/perfil	95
A.3.18. perfil/esborrar.....	98
A.3.19. perfil/editar	99
A.3.20. presentacio/contacte.....	101
A.3.21. presentacio/equip.....	103
A.3.22. presentacio/onsom.....	104
A.3.23. presentacio/projecte.....	105
A.3.24. serveis/cotreball	105
A.3.25. serveis/espais	106
A.3.26. serveis/exposicions.....	106
A.3.27. layouts/plantilla	107
A.3.28. layouts/navbar.....	108
A.3.29. errors/403.....	111
A.3.30. errors/404.....	111
A.4. Controladors.....	112
A.4.1. ActivitatController	112
A.4.2. DirectoriController	116
A.4.3. IndexController.....	119
A.4.4. PerfilController	120
A.4.5. PostController	123
A.5. Polítiques d'autorització	123
A.5.1. ActivitatPolicy.....	123
A.5.2. DirectoriPolicy	125
A.6. Notificador	126
A.6.1. RresetPasswordCat	126
A.6.2. email.blade.....	128

1. Introducció

L'origen d'aquest treball parteix de la creació d'un nou col·lectiu, que està format per cinc persones amb la voluntat d'engegar el projecte d'un espai social al districte de Sant Martí, on s'hi oferiran activitats formatives, lúdiques, culturals i de caire social. És una entitat sense ànim de lucre i amb pocs recursos inicials i necessitava un web que l'ajudés a donar-se a conèixer i a complir els seus objectius.

Amb la motivació d'aprendre sobre el desenvolupament tècnic d'un lloc web, es va plantejar la possibilitat de convertir el procés de creació del lloc web en un treball final de grau. Inicialment no es tenien gaires coneixements de programació web i per tant una part d'aquest procés va consistir en l'aprenentatge de quines són les tecnologies més adients per a resoldre les necessitats plantejades.

1.1. Problemàtica

El nou col·lectiu és de caire social i està en procés de desenvolupament i constitució. Es dedicarà principalment a la organització d'activitats en el seu local social i de manera secundària oferirà altres serveis, com la reserva de les seves sales, un espai de cotreball i un directori on puguin anunciar-se persones i entitats amb objectius semblants. Com a suport per fer tot això necessiten un lloc web que permeti connectar amb les persones que puguin estar-hi interessades i que a més ajudi a la gestió de totes aquestes tasques.

1.2. Objectius del projecte

L'objectiu principal d'aquest treball és el disseny i el desenvolupament d'un lloc web per a un nou col·lectiu. Aquesta entitat està dedicada a la gestió i dinamització d'un espai físic on s'hi realitzen activitats, cursos i serveis orientats a millorar la convivència veïnal.

Concretament, es volen aconseguir els següents objectius específics:

- Donar informació de l'entitat, del seu projecte general, dels serveis que ofereix i de les activitats concretes que s'hi realitzen

- Facilitar la gestió del projecte social, permetent que les persones responsables del mateix puguin afegir-hi informació i fer-hi canvis i que les persones usuàries de l'espai puguin interactuar-hi
- Crear el disseny i l'estructura de les pàgines que formen el lloc web
- Adaptar el lloc web a les necessitats de les persones usuàries en la mesura que sigui possible

1.3. Abast del projecte

El treball finalitzarà amb un lloc web funcional que compleixi els objectius establerts. Tot i això, aquest projecte se centrarà a la part estructural del lloc web i per tant un cop finalitzat aquest treball caldrà seguir ampliant la informació que s'hi ofereix. També cal tenir en compte les limitacions de temps per a la creació d'aquest treball que comportaran decidir un ordre de prioritats entre les funcions que ha d'oferir i possiblement deixar com a proposta de futur alguna de les funcions menys prioritàries.

2. Persones usuàries

Per definir més concretament les funcionalitats que ha de tenir el lloc web, cal tenir en compte a quin tipus de persones anirà dirigit. Les persones usuàries es poden subdividir en diferents categories segons l'ús que faran del lloc web.

2.1. Responsables de l'entitat

Aquestes persones utilitzaran el web per donar a conèixer el seu projecte, per fer promoció de les activitats i serveis que s'hi ofereixen i per millorar la gestió interna de l'entitat. En el moment d'escriure aquesta memòria, aquest grup està format per cinc persones de perfils diferents.

Totes les persones tenen entre vint i quaranta anys i tenen experiència utilitzant internet. Tot i això, pot ser necessari fer formacions internes perquè s'utilitzin eficientment els recursos del lloc web.

Les necessitats per a la gestió són:

- Poder veure els perfils de totes les persones usuàries
- Donar d'alta perfils nous
- Saber els serveis que utilitza cada persona i en quines condicions ho fa
- Poder afegir un nou servei o modificar-ne les condicions a cada perfil
- Afegir noves activitats i modificar les ja existents
- Publicar notícies sobre l'actualitat del projecte

Algunes opcions només podran realitzar-les les persones que administrin el lloc web, com ara atorgar permisos o eliminar completament una activitat.

Altres possibles utilitats poden ser:

- Veure un calendari amb totes les activitats programades a l'espai i amb informació de la gestió interna, com ara reunions programades o dates límit per a projectes
- Tenir informació estadística rellevant sobre el projecte, com ara el nombre de persones que es donen d'alta cada mes o el promig de persones apuntades per activitat

2.2. Persones usuàries de l'espai

Com és un espai dirigit a un públic ampli, les persones que poden arribar a utilitzar el web també seran molt diverses. Pel tipus d'ús que en faran, es poden diferenciar les següents subcategories: les persones que proposen activitats, les que hi assisteixen i les que fan un ús general dels serveis de l'espai.

2.2.1. Perfil de persona usuària habitual

Principalment seran persones que viuen a la zona pròxima al local. El seu interès pot ser fer vida social, tenir un punt de trobada amb altres persones i participar tant d'activitats llargues (cursos) com d'activitats puntuals (xerrades, tallers, etc.). Aquest grup serà el que faci un major ús del lloc web, ja sigui adquirint el costum de visitar-lo sovint, ja sigui entrant a través d'altres mitjans (publicitat, xarxes socials, correu electrònic...).

Un dels objectius de l'entitat social és que aquest tipus de públic sigui molt divers, incloent persones de totes les franges d'edat, de múltiples orígens socials i culturals, etc. Per tant, el lloc web ha de poder adaptar-se a unes necessitats variades:

- Diversos idiomes
- Adaptabilitat per a persones amb discapacitat visual o auditiva
- Simplicitat d'ús i múltiples rutes per a les principals funcions

2.2.2. Perfil de persona que assisteix a activitats puntuals

Aquest grup s'interessarà per activitats puntuals que hagi vist promocionades per l'entitat o per la persona o persones que les organitzin. En molts casos visitarà el web només per a una activitat concreta, amb l'objectiu de trobar-ne més informació i d'apuntar-s'hi. Per a cada activitat hi pot haver un perfil de persona específic, però si es tenen en compte totes les activitats, no hi ha un perfil clar. Per tant, les necessitats comunes seran semblants a les de la subcategoria anterior.

2.2.3. Perfil de persona que ofereix activitats

Aquestes són les persones que oferiran els cursos, tallers, xerrades, etc. Poden tenir-hi un interès econòmic o no. De les activitats se'n farà publicitat per altres vies, així que les

principals utilitats del lloc web per a aquest grup serà oferir una informació més extensa de cada activitat i permetre, a les persones interessades, apuntar-s'hi.

Alguns aspectes que és desitjable que compleixi el lloc web:

- Facilitat per veure les dates i els espais disponibles per a fer-hi activitats
- Facilitat per consultar les condicions d'ús dels serveis oferts (espai, promoció, sistema de registre, etc.)
- Sistema per fer les propostes prou senzill d'utilitzar i prou complet per a que inclogui tota la informació necessària

3. Requisits de disseny

Caldrà tenir en compte diversos aspectes per a la creació d'aquest projecte. En aquest capítol es comentaran els més destacats.

3.1. Necessitats de les persones usuàries

En el capítol anterior s'ha desenvolupat els diferents tipus de perfil de persones usuàries i les seves necessitats. A partir d'aquesta anàlisi, s'ha fet una valoració de quins són els aspectes més destacats i que caldrà tenir en compte per al disseny del lloc web. Són els següents:

- El lloc web estarà dirigit principalment a un públic extern i no necessàriament familiaritzat amb el projecte
 - Ha de tenir caràcter informatiu
 - La seva estructura ha de ser fàcil d'entendre
 - Pot ser convenient tenir un apartat de contacte
- Pot ser necessària una intranet o un espai del lloc web orientat a la seva gestió
- S'ha de poder adaptar a múltiples dispositius i mides de pantalla
- Cal que sigui accessible
- Ha de ser multilingüe

Tot i això, valorant les limitacions de temps i de coneixements, s'ha decidit que el lloc web sigui monolingüe, deixant la opció de fer-la multilingüe per a projectes posteriors de millora. De manera semblant, es treballarà en l'accessibilitat, però no es considera indispensable complir tots els requisits per a tenir un lloc web totalment adaptat, sinó només els més rellevants.

3.2. Normativa

Des del 25 de maig de 2018 és aplicable el Reglament (UE) 2016/679 del Parlament europeu y del Consell del 27 d'abril de 2016 relatiu a la protecció de les persones físiques, també conegut com a Reglament general de protecció de dades (RGPD) [1]. És una normativa europea directament aplicable (no requereix que s'adaptin les lleis estatals perquè s'hagi de complir). Aquest reglament se superposa amb l'anterior Llei Orgànica 15/1999, del

13 de desembre [2], de protecció de dades de caràcter personal (LOPD) i el Reial Decret 1720/2007, del 21 de desembre [3], conegut com el Reglament de desenvolupament de la LOPD, que també cal que el lloc web compleixi.

Com d'entrada no hi haurà comerç electrònic ni publicitat a través d'aquest lloc, no cal aplicar la Llei 34/2002, de l'11 de juliol [4], de serveis de la Societat de la Informació i Comerç Electrònic (LSSICE o LSSI), però caldrà tenir-la en compte si en un futur es decideix integrar elements de comerç o de publicitat en el lloc web i també si es decideix enviar correus electrònics amb fins publicitaris.

Els aspectes relacionats amb la normativa aplicable són els següents:

- Cal establir clarament quina és la base legal dels tractaments de dades realitzats.
- Hi ha d'haver un consentiment explícit per part de les persones usuàries per al tractament de les seves dades. Cal que el text perquè es pugui donar aquest consentiment sigui clar i fàcil d'entendre. Això significa que tots els formularis en els que les persones usuàries envïin informació personal han d'incloure una casella de verificació del consentiment, junt amb un text explicatiu.
- La informació legal relacionada amb les dades personals ha de ser de fàcil accés.
- Els drets d'ús de les dades personals s'han de poder revocar de manera senzilla, com ara a través d'un correu electrònic. Els mecanismes establerts han de ser clarament identificats.
- El consentiment de les persones menors de setze anys requereix també el consentiment dels seus pares o de les persones que en tinguin la tutoria legal.
- Cal protegir les dades personals de les persones usuàries, garantint la integritat de la informació i establint mecanismes per a la seva recuperació si fos necessari.
- Hi ha d'haver un registre de les activitats de relacionades amb el tractament de dades.
- Si s'utilitzen altres empreses per a activitats relacionades amb el tractament de dades, cal assegurar-se que també compleixin la normativa.

3.3. Requisits de l'entitat

El lloc web formarà part de la imatge de l'entitat que hi ha al darrere. Per això cal tenir en compte algunes de les característiques d'aquesta organització, així com les demandes que tingui:

- L'idioma principal ha de ser el català
- S'ha de fer un ús del llenguatge no sexista

- Es dóna molta importància als principis socials, cal evitar les imatges i les expressions que no respectin aquests principis
- S'utilitzarà el logotip existent i per tant cal tenir-lo en compte per a la tria de colors i altres aspectes estètics

3.4. Aspectes tècnics

La tria dels aspectes tècnics es desenvoluparà en els propers capítols. En aquest apartat se senyalen els primers elements que cal tenir en compte.

- S'utilitzaran llenguatges de programació com HTML, CSS i PHP
- Caldrà utilitzar una base de dades
- S'hauran de generar relacions entre les taules de la base de dades i un sistema visual per interactuar-hi
- Caldrà tenir en compte la seguretat del lloc web davant de possibles atacs des d'internet

4. Eines de programació

La programació d'un lloc web es pot fer de diverses maneres. Seria possible crear tot el codi des de zero utilitzant els llenguatges html, css, javascript, php i sql. Això implicaria crear totes les funcions i tot el disseny. Aquest sistema té alguns desavantatges importants:

- Comporta molta feina repetitiva.
- El risc de cometre errors és elevat.
- És difícil que es pensi en totes les possibilitats a nivell de seguretat i per tant, és més probable que el disseny creat tingui forats de seguretat greus.

Com a alternativa en el món de programació web fa temps que existeixen els *frameworks* i les llibreries. Un *framework* és una àrea de treball per programar, és a dir, un conjunt d'eines preparades per començar a desenvolupar la programació d'un lloc web i a més fer-ho de manera fàcil i segura. Aquesta opció és molt més utilitzada perquè té nombrosos avantatges:

- Significa aprofitar feina que ja està feta i per tant es redueix el temps necessari per programar.
- És més segur, ja que el codi utilitzat ha estat provat durant més temps, s'han tingut en compte més factors i, si les llibreries estan actualitzades, aprofiten les innovacions que s'han anat creant en els diferents llenguatges utilitzats.
- El codi resulta més clar i facilita la tasca de treballar en equip i la de que altres persones puguin agafar el relleu en el manteniment i programació del lloc web. Això és gràcies a que se segueixen uns estàndards marcats per a la creació del codi amb un *framework* en particular.

El principal desavantatge d'utilitzar aquesta opció és que són necessaris més coneixements previs per a començar. I a més, aquests coneixements canvien entre diferents *frameworks* i diferents llibreries. Tot i això, calen menys coneixements específics de cada llenguatge de programació utilitzat en comparació a la opció de crear tot el codi des de zero.

4.1. Laravel

Existeixen múltiples *frameworks* actualment, molts d'ells de codi obert. El que s'ha triat per a aquest projecte és *Laravel 5.6* [5].

La principal raó per la que s'ha triat *Laravel* com a *framework* a utilitzar és que és un dels més populars actualment i per tant és fàcil trobar recursos relacionats, tant tutorials com llibreries basades en *Laravel*. A més és senzill d'utilitzar i alhora permet moltes opcions.

Laravel és un *framework* basat en una arquitectura Model-Vista-Controlador (*Model-View-Controller*, MVC), és a dir, en la que se separen l'aplicació i la vista. La utilització d'aquest *framework* afecta a com es generarà tot el projecte del lloc web.

4.2. Panell d'administració

Per tal de controlar el lloc web, pot ser útil tenir un panell de control des d'on gestionar diferents aspectes: persones usuàries, rols, permisos, articles, activitats programades... Per cada aspecte, s'han de poder fer les quatre funcions bàsiques: crear, llegir, actualitzar (modificar) i guardar, també conegudes com a *CRUD* (de l'anglès, *create, read, update and delete*).

A l'hora de generar aquest panell de control, es pot fer de diverses maneres. Una opció és crear tot el panell de control des de zero. Això permetria crear directament un panell que s'adapti totalment a les necessitats del projecte, però també comporta molta feina repetitiva. Es pot reduir part d'aquesta feina utilitzant alguns dels paquets preparats per a *Laravel* de manera modular. Per exemple, es pot utilitzar un paquet generador de *CRUD* o un paquet amb algunes vistes predefinides o un sistema de registre de noves persones usuàries que hi inclou també l'administració, etc.

D'altra banda, també existeixen múltiples paquets per a *Laravel* que inclouen tot el sistema d'administració. Cada paquet és diferent, alguns d'aquests paquets són més rígids, altres permeten adaptar-los més a les necessitats de cada projecte i poden incloure funcionalitats diferents. També cal tenir en compte que alguns són de codi obert, que permet l'ús gratuït, mentre altres són de pagament.

Per a aquest projecte s'ha triat la opció d'utilitzar *Voyager 1.1* [6], un paquet de codi obert per a *Laravel*. Inclou un panell de control fàcil d'utilitzar i amb moltes funcionalitats. Amb aquest paquet se simplifica la tasca de crear taules noves i generar relacions entre taules. D'entrada crea algunes taules a la base de dades triada que són freqüents a molts llocs web, inclòs el d'aquest projecte. Si es considera necessari, s'hi inclou una funció generadora de menús, però per a aquest projecte no s'utilitzarà. També es pot fer compatible amb *Google Analytics* per veure les estadístiques d'ús del lloc web. Finalment, també inclou un sistema de gestió d'arxius audiovisuals.

4.3. Bootstrap

Tal com s'indica a la seva pàgina principal, *Bootstrap* [7] és un conjunt d'eines de codi obert per al desenvolupament amb html, css i javascript i està basat en jQuery.

Una de les característiques de disseny més importants és que el lloc web s'ha de poder consultar còmodament tant en un ordinador de sobretaula com a través de mòbils i tauletes. Per aconseguir això, és interessant la filosofia de *Bootstrap*, dissenyar primer pel mòbil. Les principals característiques de *Bootstrap* 3.3 [8] útils per a aquest projecte són:

- Adapta la visualització segons la mida de la pantalla que s'estigui utilitzant
- Els seus arxius són molt utilitzats i per tant és probable que ja estiguin en la memòria *cache* del client (aparell receptor)
- Facilita la creació de graelles on incloure elements visuals

Amb aquesta eina en el projecte es crearà el menú principal de navegació. Això permetrà que per una banda es mostri en horitzontal en pantalles d'ordinador mitjanes i grans i d'altra banda en mòbils i tauletes només apareguin la icona de la pàgina d'inici i una icona desplegable amb la resta del menú.

També s'utilitzarà per millorar la visualització general de la web, situant els elements entre sí i canviant-los de mida i de posició en funció de la mida de pantalla. Serà especialment útil quan es vulguin mostrar múltiples elements del mateix tipus (com ara imatges o quadres de text).

4.4. Formulari

Dins del lloc web es vol incloure un formulari de contacte per a que qualsevol persona que vulgui enviar un missatge pugui fer-ho directament a través de la web. Per crear un formulari cal tenir en compte d'una banda la seguretat de la informació que s'envia i a més la possibilitat que bots l'utilitzin per enviar massivament correus de publicitat. Per simplificar la programació de tot això s'ha triat utilitzar un servei de missatges amb un formulari pre-dissenyat.

Concretament, es farà servir *Formcarry* [9]. Té diferents tarifes segons el servei que es vulgui. Amb la versió gratuïta es pot crear un formulari per compte i que s'hi enviïn fins a cent missatges mensuals. Tenint en compte les necessitats inicials d'aquest lloc web aquest servei les cobrirà amb un ampli marge (les previsions són que no arribin més de deu missatges al mes per aquesta via).

4.5. Sistema de registre i verificació de persones usuàries

El paquet *Voyager* inclou una part de la creació d'un sistema de registre per a les persones usuàries, que és la que es correspon amb l'estructura interna i amb l'administració des del panell de control. Per tenir el sistema complet, cal afegir les vistes corresponents i les seves rutes. Per facilitar també aquesta part, *Laravel* pot generar-les amb el paquet *Auth* [10] que s'hi inclou per defecte. A partir d'aquí, només cal personalitzar tots els elements que es considerin necessaris.

En aquests paquets hi ha una pàgina de registre, una d'inici de sessió (*login*), una per a restablir la contrasenya (enviant un correu electrònic), una per fer el canvi de contrasenya un cop s'ha restablert i una vista amb el missatge de confirmació de l'inici de sessió. Com a funcions destacades, per una banda es pot gestionar la creació, modificació i destrucció de perfils des del panell d'administració i per altra banda qualsevol persona usuària pot crear un nou perfil i iniciar i tancar sessió. També s'hi inclouen alguns mètodes que simplifiquen tasques com filtrar el tipus de persones usuàries que poden veure una pàgina o que una part del menú principal canviï segons si una persona està registrada o no.

A més del benefici que suposa estalviar feina a través de les eines utilitzades, en aquest apartat és especialment rellevant l'avantatge que les operacions de guardar i recuperar contrasenyes segueixin uns protocols de seguretat contrastats. A través dels paquets anomenats, durant el registre es demana una contrasenya i la seva verificació que han de coincidir. I per al guardat a la base de dades, la contrasenya es codifica, de manera que ni tan sols les persones que administrin el lloc web puguin veure les contrasenyes de les persones usuàries.

5. Estructura

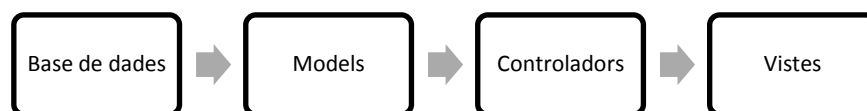
En aquest capítol s'explicarà amb quina estructura funcionarà el lloc web. D'una banda, l'estructura interna, és a dir, el sistema d'arxius i carpetes, i d'altra banda, l'estructura externa, com s'organitzen les diferents pàgines.

Per a aquesta segona part es diferencia entre les pàgines que podran veure la major part de persones usuàries i el panell d'administració, al que només podran accedir-hi persones amb els permisos necessaris.

5.1. Estructura de carpetes

Tal com s'ha explicat en el capítol anterior, s'ha decidit treballar amb *Laravel*, que té una estructura de carpetes per defecte que cal mantenir. Sobre aquesta estructura s'hi han fet algunes modificacions per adaptar-la a les necessitats d'aquest projecte en particular.

El model MVC consisteix en separar el codi en models, vistes i controladors [11]. Els models es relacionen amb la base de dades. Inclouen les funcions que afecten directament a l'estructura de la base de dades. Les vistes corresponen amb la part més visible del web, on es crea el disseny visual. Es connecten amb la resta del codi a través dels controladors.



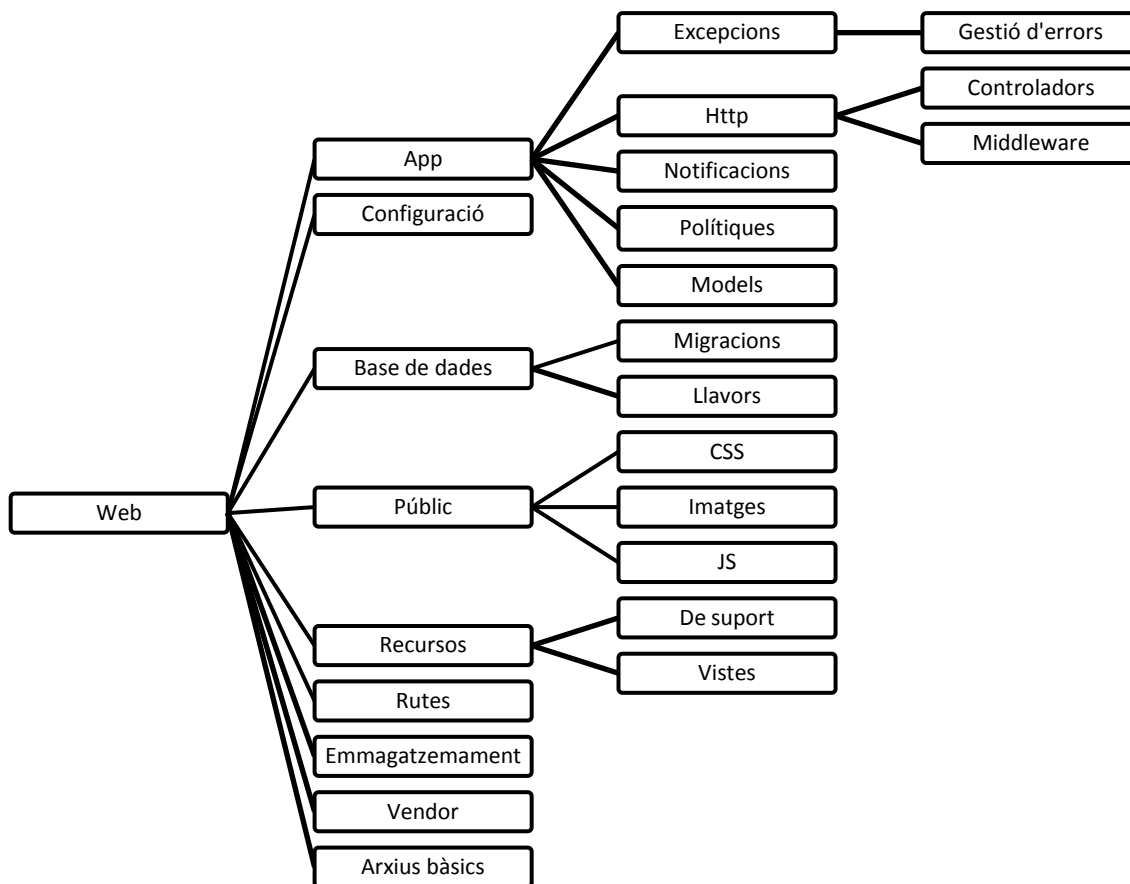
Imatge 5.1: Esquema de funcionament d'un model MVC

L'estructura de carpetes de *Laravel* és més complexa, tot i que segueix un planejament MVC. A la imatge 5.2 es pot veure un esquema simplificat del sistema de carpetes que s'ha utilitzat. No hi apareixen totes les carpetes de l'aplicació sinó només les que s'han considerat més rellevants per a l'explicació del funcionament d'aquesta estructura.

A la carpeta *App* hi ha la part de codi que es correspon amb la lògica utilitzada en l'ús comú del lloc web. És on hi ha els controladors (dins d'una subcarpeta) i els models. A més també hi ha els arxius de gestió d'errors de l'aplicació, els d'enviament de notifikacions per correu

electrònic i les polítiques sobre a quines pàgines pot accedir cada persona usuària. Junt amb els controladors també hi ha els *middlewares* [12], que són els arxius on hi ha les funcions que gestionen quines vistes poden veure els diferents tipus de persones usuàries segons els permisos que tinguin.

La carpeta de configuració conté tots els arxius relacionats amb la configuració de l'aplicació, on es detallen coses com l'idioma del web, els paquets, quina base de dades s'utilitza, etc. Mentre que a la de base de dades s'hi guarden els arxius amb els que es creen les bases de dades (les migracions) i els arxius amb els que s'omplen amb informació preexistent.



Imatge 5.2: Esquema simplificat de l'estructura de carpetes

A la carpeta *Públic* hi ha els recursos que s'utilitzen des de les vistes, és a dir, els arxius d'estil CSS, les imatges que hi apareixeran i els arxius de javascript, que contenen les funcions que s'executen en el dispositiu client (i no en el servidor). Les vistes estan a la carpeta de recursos, junt amb altres carpetes de suport.

Les rutes són una part important d'aquesta estructura perquè són les que ajuden a connectar les vistes amb els seus controladors i és des d'on es creen les direccions que apareixen en els navegadors. A l'emmagatzemament hi ha per una banda informació guardada addicional a la base de dades i per altra banda informació de suport, com les sessions existents.

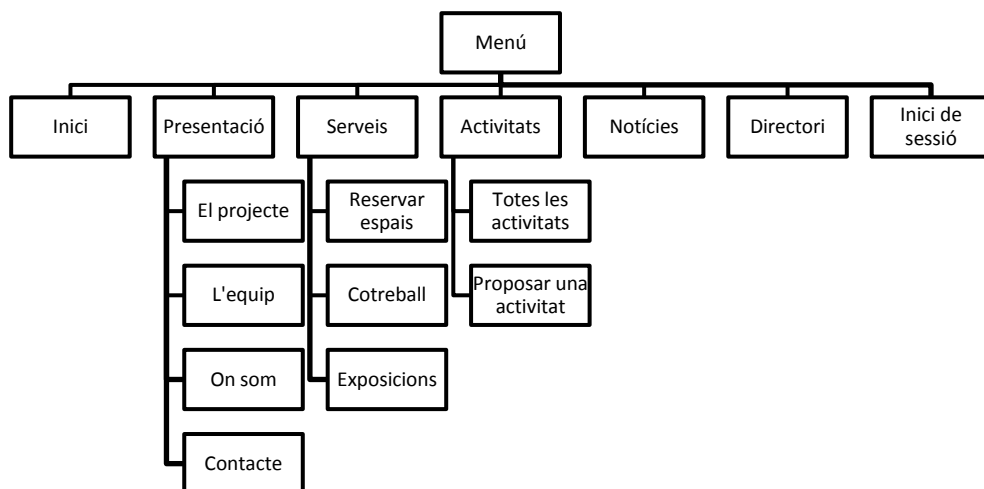
A la carpeta *Vendor* hi ha els arxius propis de *Laravel* i dels paquets instal·lats que formen l'estructura bàsica del seu funcionament i que per tant no convé no modificar. Molts dels arxius d'aquesta part es poden sobre escriure, creant un altre arxiu a altres carpetes, que s'utilitzarà en lloc de l'arxiu original. Tot i això no està recomanat esborrar els arxius originals i per tant s'han mantingut tal com eren en el moment de la seva instal·lació.

Finalment, estan els arxius bàsics, on destaca l'arxiu *.env*, on s'hi indica la informació d'entorn rellevant: el nom de l'aplicació, la informació del servidor, la connexió amb la base de dades, la configuració dels correus electrònics, etc.

En capítols posteriors es desenvoluparà el funcionament dels principals elements d'aquest esquema i la relació entre ells.

5.2. Menú principal

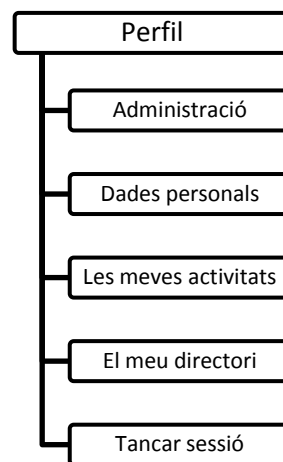
A totes les pàgines del lloc web hi haurà un menú principal comú. En el menú per defecte (que és el que apareix quan no s'ha iniciat sessió) s'hi inclouen els apartats que es poden veure a la imatge 5.3.



Imatge 5.3: esquema del menú principal (sense iniciar sessió)

S'ha volgut reduir el nombre d'apartats al mínim possible mantenint la utilitat del menú. Quan es visualitza en una pantalla d'ordinador es veuen tots els apartats sense donar sensació d'excés. Tots els apartats que tenen subapartats són desplegable, de manera que només es vegin quan són necessaris.

D'altra banda, a través d'una tauleta o un mòbil d'entrada només es veu l'opció d'inici (representada amb una icona) i la opció de desplegar el menú (també mitjançant una icona), de manera que es pugui veure la pàgina d'inici més directament i el menú només aparegui quan la persona usuària ho requereixi.

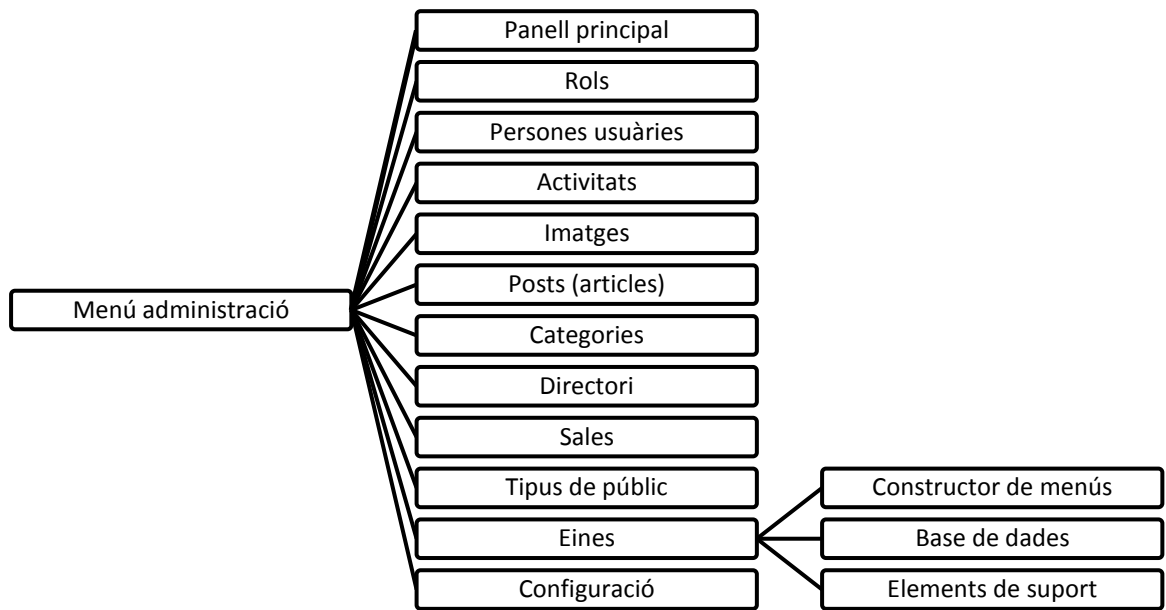


Imatge 5.4: esquema de l'apartat del perfil (amb la sessió iniciada)

El menú tindrà alguns canvis segons qui l'estigui veient. Canviaria l'apartat *inici de sessió* per un accés al perfil propi, que es pot veure a la imatge 5.4. Serà també un desplegable que inclourà altres subapartats que dependran del tipus de perfil. A la majoria de persones registrades els apareixeran opcions perquè puguin modificar les seves dades, veure les activitats de les que són responsables i aquelles a les que s'han apuntat. El subapartat *Administració* només servirà visible per a les persones que puguin tenir accés al panell d'administració.

5.3. Estructura del panell d'administració

El panell d'administració segueix l'estructura per defecte de *Voyager*, però amb més apartats, que es corresponen a les taules de la base de dades que s'han afegit. En aquest cas el menú de navegació és lateral i una de les opcions també és desplegable.



Imatge 5.5: Esquema del menú del panell d'administració

La majoria de les opcions del menú obren un llistat amb tots els elements d'aquell tipus, per exemple *Posts* obre un llistat amb tots els articles que hi ha a la base de dades. Des d'aquesta pàgina del llistat es pot accedir a les opcions de creació de nous elements i modificació o esborrat dels existents.

El panell principal mostra quantes persones usuàries i quants articles hi ha a la base de dades, amb un enllaç per a cada pàgina de gestió respectivament. Si es vincula amb *Google Analytics* [13], aquí es podran veure les estadístiques relacionades amb l'ús del lloc web.

Dins del camp *Eines* hi ha l'opció que permet fer la gestió de les bases de dades. A la pàgina que obre es presenta una llista amb la majoria de les taules de la base de dades, les que s'han considerat rellevants. Des d'aquí es pot accedir a les opcions de veure, modificar o eliminar cada taula i a més també a les mateixes opcions respecte als *BREAD*, que es corresponen amb les pàgines de gestió de cada taula (les mateixes que apareixen en el menú). Amb aquesta última opció es poden crear les relacions entre taules.

El constructor de menús, en aquest projecte, només s'ha utilitzat per triar quins elements i per quin ordre han de ser visibles en el menú del panell d'administració. En els elements de suport s'hi poden trobar recursos útils com les icones de *Voyager*, que s'utilitzen en la visualització d'aquest menú, o com el llistat d'errors que s'han produït en el lloc web.

Finalment, a l'apartat de configuració s'hi poden definir diferents camps rellevants de l'aplicació per poder fer-hi referència a través de *Voyager*, com ara el nom i una petita

descripció del lloc web. Està separada en dues pestanyes dins de la pàgina, una per al lloc web i una per al panell d'administració.

A més del menú principal, a la part superior hi ha una icona corresponent a l'avatar de la persona usuària. Des d'aquesta es pot obrir un desplegable amb les opcions de veure el perfil, tornar a la pàgina principal del web o tancar la sessió.

5.4. Capçalera i peu de pàgina

A més del menú principal de navegació, hi haurà dos elements més que estaran presents a totes les pàgines del lloc web, que són la capçalera i el peu de pàgina. Tots dos també tindran alguns elements de navegació.

La capçalera estarà formada pel logotip i per enllaços a les xarxes socials. En una pantalla d'ordinador es veurà el logotip gran i a l'esquerra, mentre que a la dreta hi haurà les icones de les xarxes socials que decideixi la cooperativa, que d'entrada seran a Twitter, Facebook i una adreça electrònica.

En el peu de pàgina hi haurà diversos enllaços. D'una banda, s'enllaçarà a una pàgina on estiguin els avisos legals que apareixen en el lloc web. També, quan existeixin, hi haurà un enllaç a la llicència d'ús o drets d'autoria del lloc web. I finalment estaran les dades de contacte de l'entitat.

6. Base de dades

Per poder gestionar la informació serà necessari l'ús de bases de dades. S'ha triat *MySQL* [14] com a gestor perquè és un dels més utilitzats i per tant és fàcil trobar informació sobre el seu ús, els errors que s'hi puguin produir i programari que sigui compatible amb aquest gestor. Concretament *Laravel*, tot i que permet utilitzar altres tipus de bases de dades, per defecte està preparat per fer servir *MySQL* [15]. Aquest gestor de base de dades és de codi obert i té dos tipus de llicències, una pública de lliure distribució i una comercial que inclou alguns serveis addicionals.

La base de dades està allotjada en el servidor i s'hi accedeix a través de la consola i de les comandes en el propi lloc web. Utilitzant *Laravel* i *Voyager* no cal escriure codi SQL, ja que la comunicació amb la base de dades està generada per defecte.

Quan s'instal·la *Voyager* es creen algunes taules a la base de dades, concretament les més rellevants són:

- Persones usuàries
- Rols
- Permisos
- Pàgines
- Posts (articles)
- Categories
- Traduccions
- Menús
- Configuració
- Migracions

A més, també crea algunes bases complementàries per guardar relacions entre taules i per generar les funcions BREAD (sigles en anglès, *browse, read, edit, add and delete*, és a dir, cercar, llegir, editar, afegir i esborrar) en el panell d'administració.

Algunes d'aquestes taules serveixen perquè funcioni el panell de *Voyager* i per tant són necessàries, com la taula de migracions (necessària per poder crear noves taules a la base de dades) o les taules de menús (on està el menú del panell d'administració). Però no es faran canvis directament en aquestes taules.

En canvi, altres taules requeriran una adaptació perquè siguin útils per al projecte. A més, també caldrà crear-ne de noves. Això es farà a través del panell d'administració, que a més

de ser una manera més simple de crear taules, també crea les vistes necessàries del panell d'administració amb les funcions BREAD.

En els apartats següents s'especifiquen tant les taules noves com les taules que es modificaran.

6.1. Taula de persones usuàries

La taula de persones usuàries serà la mateixa que *Voyager* genera per defecte, amb alguns camps addicionals. El número identificatiu de cada fila s'autoincrementa i no pot ser nul. A la columna rol, s'hi guarda el número identificatiu del rol corresponent a aquesta persona. L'adreça electrònica i la contrasenya no poden ser nul·les perquè s'utilitzen per iniciar sessió. A més l'adreça electrònica també serveix per comprovar que no existeix cap altre perfil amb la mateixa adreça i per recuperar la contrasenya en cas que sigui necessari. També és obligatori el camp *nom* perquè s'utilitza en diferents punts del lloc web.

El camp *token recordatori* s'utilitza en cas que la persona usuària obli la contrasenya i demani restablir-la. A *configuració* s'hi pot guardar informació que no estigui en altres camps, com l'idioma que prefereix veure per defecte si el lloc web és multilingüe. Els camps de data i hora de creació i modificació serviran per controlar en cas que sigui necessària la supervisió de l'administració del web.

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Número identificatiu	id	int	10	no
Rol	role_id	int	10	sí
Nom	name	varchar	255	no
Adreça electrònica	email	varchar	255	no
Avatar	avatar	varchar	255	sí
Contrasenya	password	varchar	255	no
<i>Token</i> recordatori	remember_token	varchar	100	sí
Configuració	settings	text		sí
Data i hora de creació	created_at	timestamp		sí
Data i hora de modificació	updated_at	timestamp		sí

Taula 6.1: taula de persones usuàries

Les dades poden ser de tipus *int* (número enter), *varchar* (caràcters del sistema de codificació UTF8), *timestamp* (marca temporal, en aquest cas referent a la data i l'hora), text

(text amb una extensió màxima de 65 kB). A la taula 6.1 hi ha la llista dels camps de la taula, on les longituds màximes de cada camp estan indicades en bytes. La columna *Permet nul* es refereix a la possibilitat de deixar buit aquest camp.

6.2. Rols, permisos i relacions entre taules

S'ha considerat necessari crear una estructura de rols i permisos, de manera que cada persona usuària tingui un i només un rol i cada rol pugui tenir múltiples permisos. Cada permís es refereix a una acció concreta, per exemple crear un nou article o esborrar una activitat. Perquè el lloc web funcioni correctament i algú pugui fer-ne el manteniment, cal que com a mínim hi hagi un perfil amb el rol d'administració, que té tots els permisos possibles.

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Número identificatiu	id	int	10	no
Nom	name	varchar	255	no
Nom per a mostrar	display_name	varchar	255	no
Data i hora de creació	created_at	timestamp		sí
Data i hora de modificació	updated_at	timestamp		sí

Taula 6.2: taula de rols

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Número identificatiu	id	int	10	no
Clau	key	varchar	255	no
Nom de la taula	table_name	varchar	255	sí
Data i hora de creació	created_at	timestamp		sí
Data i hora de modificació	updated_at	timestamp		sí

Taula 6.3: taula de permisos

Els permisos es creen en blocs de funcions BREAD. Cada bloc està relacionat amb una taula. Per aquesta raó la taula de permisos té les columnes *nom de la taula*, que indica la taula amb que es relaciona cada permís, i *clau*, que especifica a quina funció es refereix. A més també hi ha alguns permisos relacionats amb funcions genèriques no vinculades a cap taula, per això tots els permisos han de tenir clau però no cal que tots tinguin un nom de taula associada.

La taula de relacions entre rols i permisos guarda els números identificatius de les files de cada una de les taules que es volen relacionar. Generar aquesta taula apart de les altres dues permet crear relacions més complexes en les que un rol pot tenir múltiples permisos i un permís pot vincular-se a més d'un rol.

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Permís	permission_id	int	10	no
Rol	role_id	int	10	no

Taula 6.4: taula de relacions entre permisos i rols

6.3. Taula de categories

Les categories serveixen per classificar i per fer cerques. S'utilitzaran en els articles, les activitats i el directori. En aquesta taula és on es guarden i on se'n poden afegir de noves. El camp *categoria pare* permetria crear relacions de dependència entre categories si es considera convenient, tot i que en aquest projecte no s'utilitzaran. Totes les columnes, inclosa aquesta, s'han generat automàticament per *Voyager*, s'ha decidit mantenir-la per si en un futur es vol utilitzar. L'*ordre* indica si la categoria té alguna categoria superior, per tant en aquest cas serà 1 en tots els casos.

L'*slug* és la part de la URL (*Uniform Resource Locator*, localitzador uniforme de recursos) que es correspon amb cada categoria en particular. Guardar aquesta informació és necessari per fer URL *amigables*, és a dir, que siguin comprensibles i informin del contingut de cada pàgina.

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Número identificatiu	id	int	10	no
Categororia <i>pare</i>	parent_id	int	10	sí
Ordre	order	int	11	no
Nom	name	varchar	255	no
Slug	slug	varchar	255	no
Data i hora de creació	created_at	timestamp		sí
Data i hora de modificació	updated_at	timestamp		sí

Taula 6.5: taula de categories

6.4. Taules d'activitats i de tipus de públic

Cal crear una taula d'activitats on hi hagi la informació de les activitats que es fan des de la organització. El títol de l'activitat és obligatori. Les columnes *sales* i *tipus de públic* són per relacionar-les amb altres taules, la primera amb la de sales i espais i la segona amb la de tipus de públic. Són per saber on es farà l'activitat i a qui està dirigida.

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Número identificatiu	id	int	10	no
Títol	titol	varchar	255	no
Data d'inici	data_inici	date		sí
Hora d'inici	hora_inici	time		sí
Data de finalització	data_final	date		sí
Hora de finalització	hora_final	time		sí
Sala	sala	int	10	sí
Número de places màximes	places_max	int	11	sí
Resum	resum	text		sí
Descripció llarga	descr_llarga	mediumtext		sí
Data límit per inscriure's	data_limit_inscr	date		sí
Imatge	imatge	varchar	255	sí
Tipus d'activitat	tipus_act	varchar	255	sí
Tipus de públic a qui va dirigit	tipus_public	int	10	sí
Preu	preu	varchar	255	sí
Taquilla inversa (sí/no)	taquilla_inv	varchar	60	sí
Slug	slug	varchar	255	sí
Etiqueta títol	titol_seo	varchar	255	sí
Paraules clau meta	meta_keywords	text		sí
Estat	estat	varchar	255	sí
Horari	horari	text		sí
Direcció	direccio	text		sí
Destacada (sí/no)	destacada	varchar		sí
Data i hora de creació	created_at	timestamp		sí
Data i hora de modificació	updated_at	timestamp		sí
Data i hora d'esborrat	deleted_at	timestamp		sí

Taula 6.6: taula d'activitats

A més de la informació de quan s'ha creat i quan s'ha modificat cada activitat, s'ha decidit afegir la data i l'hora d'esborrat, que permet ocultar la informació, però mantenir-la guardada. D'aquesta manera es vol prevenir possibles esborrats accidentals. També es guardarà la

data i l'hora de l'inici i del final de l'activitat, així es pot donar aquesta informació destacada i diferenciada de la descripció i a més, s'utilitzaran aquestes dades en el lloc web per programar quan temps ha d'estar visible una determinada activitat.

Entre els tipus de dades, a més de les explicades a l'apartat [6.1](#), hi ha *date* (data), *time* (hora) i *mediumtext* (text amb una extensió màxima de 16 MB). El *resum* és de tipus text perquè s'hi pugui escriure un paràgraf, però per a poder redactar una descripció més llarga s'ha considerat que el tipus *text* podria ser insuficient en alguns casos i que per tant per a aquest camp és preferible utilitzar el format *mediumtext*, que té més capacitat.

El preu és de tipus *varchar* perquè s'hi pugui escriure una frase curta, en comptes de només un número. Per exemple, es podria escriure que és gratuïta, donar un interval de preus o posar preus amb descomptes per a alguns col·lectius. L'opció de *taquilla inversa*, amb dues opcions, sí o no, s'ha afegit com a requeriment per part de l'organització per a la que es fa el lloc web.

L'*etiqueta títol* es correspon amb l'etiqueta *<title>* en el codi html, que apareixerà a la pestanya del navegador i a més és utilitzat pels cercadors de contingut a internet i pels navegadors quan es guarda la pàgina. Les *paraules clau meta* també s'utilitzen en els cercadors de contingut i ajuden a millorar el posicionament de la pàgina.

Al camp d'estat s'hi guardarà l'estat de la publicació de l'activitat, que pot ser publicada, pendent de revisió o en esborrany. L'horari s'utilitzarà només en cas que l'activitat sigui un curs. És de tipus text perquè s'hi pugui escriure qualsevol possible horari. La direcció per defecte serà la del local, però es guarda amb un camp diferenciat per a modificar-la si és necessari. L'activitat es podrà indicar com a destacada en aquest camp, que admet dues possibilitats, sí o no. Servirà per a saber quina activitat cal destacar a la pàgina principal.

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Número identificatiu	id	int	10	no
Tipus	tipus	varchar	255	no
Data i hora de creació	created_at	timestamp		sí
Data i hora de modificació	updated_at	timestamp		sí
Data i hora d'esborrat	deleted_at	timestamp		sí

Taula 6.7: taula de tipus de públic

S'ha decidit crear una taula de tipus de públic on s'hi guardin les possibilitats de públic a qui pot anar dirigida una activitat. Per exemple, persones majors d'edat o membres d'un

determinat col·lectiu. Tenir aquestes opcions en una taula facilita que pugui triar-se una de les ja existents o bé afegir-ne més si és convenient.

També s'han creat dues taules de suport, una per relacionar les activitats amb les persones usuàries responsables de cada activitat i una altra per la relació amb les assistents. En tots dos casos són taules amb només dues columnes: *activitats_id* i *users_id*, on estan els respectius números identificatius. Les persones responsables de l'activitat podran modificar-la o esborrar-la. La taula d'assistents és perquè les persones usuàries que vulguin puguin inscriure's a l'activitat. En alguns casos això pot ser un requisit obligatori per assistir-hi (en aquest cas s'indicaria a la descripció de la taula d'activitats).

6.5. Taula d'articles

Els articles o *posts* aniran guardats a la taula generada per *Voyager* amb aquesta finalitat. Amb les columnes per defecte serà suficient per a l'objectiu d'aquest apartat, per tant, no s'hi faran modificacions. Les columnes d'autoria i de categoria són per als identificadors de les taules de persones usuàries i de categories respectivament.

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Número identificatiu	id	int	10	no
Autoria	author_id	int	11	no
Categoria	category_id	int	11	sí
Títol	title	varchar	255	no
Etiqueta títol	seo_title	varchar	255	sí
Extracte	excerpt	text		sí
Cos	body	text		no
Imatge	image	varchar	255	sí
Slug	slug	varchar	255	no
Descripció meta	meta_description	text		sí
Paraules clau meta	meta_keywords	text		sí
Estat	status	enum('published', 'draft', 'pending')		no
Destacat	featured	tinyint	1	no
Data i hora de creació	created_at	timestamp		sí
Data i hora de modificació	updated_at	timestamp		sí
Data i hora d'esborrat	deleted_at	timestamp		sí

Taula 6.8: taula d'articles

Hi ha dues columnes de títol, la primera és per al títol que es veurà dins de la pàgina, encapçalant el text de l'article. El segon és per a l'etiqueta *<title>*, com en el cas de les activitats. També hi ha les columnes per a la *descripció* i les *paraules clau meta* que utilitzen els cercadors d'internet.

El text de l'article està separat en dues parts, un extracte breu i el cos. La imatge permet associar una imatge a l'article. L'estat és del tipus enumeració (*enum*) i admet tres possibilitats: publicat, esborrany i pendent. Finalment, l'apartat *destacat* indica si un article en particular ha d'estar destacat o no. Admet dues opcions, 0 si no es destaca i 1 si ho fa.

6.6. Taula de sales i espais

Un dels apartats de la web inclourà les sales i espais disponibles en el local per fer-hi activitats. A més el nom de les sales també és necessari a les pàgines de les activitats, per saber on es farà cada una. A la taula hi ha un apartat per escriure-hi una petita descripció, un per incloure una imatge de la sala i el de la capacitat, és a dir, el número màxim de persones que està previst que puguin estar a cada espai.

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Número identificatiu	id	int	10	no
Nom	nom	varchar	255	no
Descripció	descripcio	text		sí
Imatge	imatge	varchar	255	sí
Capacitat	capacitat	int	5	sí
Data i hora de creació	created_at	timestamp		sí
Data i hora de modificació	updated_at	timestamp		sí
Data i hora d'esborrat	deleted_at	timestamp		sí

Taula 6.9: taula de sales i espais

6.7. Taula per al directori

Per al directori s'ha cregut necessari crear una taula pròpia. Es podria haver fet el directori a través de la taula de persones usuàries, però s'ha considerat que separar-les ajuda a la claredat i a més, així és més fàcil afegir persones o entitats que estiguin només interessades

en aparèixer al directori però no volen tenir un perfil propi. Per fer-ho d'aquesta manera, s'ompliria el camp *nom*, però el d'*identificatiu de persones usuàries* quedaria buit.

El camp *nom* és obligatori i és el que es mostraria en primer lloc en el directori. En canvi, l'identificador de persones no és obligatori i serveix per a que les persones que han creat la seva entrada del directori puguin mantenir-ne el control i modificar-les o eliminar-les si ho volen.

Hi haurà dos tipus d'entrades, les de persones físiques i les d'entitats, si és una o altra es guardarà a la columna corresponent. També hi ha espai per a la informació de contacte: lloc web, adreça electrònica i postal. Aquesta última és del tipus *text* per a que pugui ser una mica més extensa. S'hi podrà guardar una petita descripció en un camp també tipus *text* i una imatge, que pot ser la mateixa que l'avatar del perfil o no.

Columna	Nom columna	Tipus de dades	Longitud màxima	Permet nul
Número identificatiu	id	int	10	no
Nom	nom	varchar	255	no
Identificatiu de la persona usuària	user_id	int	10	sí
Tipus (persona/entitat)	tipus	varchar	255	sí
Web	web	varchar	255	sí
Adreça electrònica	email	varchar	255	sí
Adreça postal	adrecapostal	text		sí
Descripció	descripcio	text		sí
Imatge	imatge	varchar	255	sí
Categories	categoria_id	int	10	sí
Estat (actiu/inactiu)	estat	varchar	255	sí
Data i hora de creació	created_at	timestamp		sí
Data i hora de modificació	updated_at	timestamp		sí
Data i hora d'esborrat	deleted_at	timestamp		sí

Taula 6.10: taula del directori

S'hi ha afegit el camp *categories* per ajudar a fer cerques. En aquest camp s'hi guarda l'identificador de la taula corresponent. I finalment a l'*estat* s'hi admeten dues opcions: actiu i inactiu, de manera que no calgui esborrar una entrada del directori perquè deixi de ser visible.

6.8. Altres taules

Voyager genera dues taules que no s'utilitzaran de moment, que són la de *pàgines* i la de *traduccions*. Si es fa un lloc web multilingüe poden ser molt útils, però en aquest projecte s'ha decidit fer un disseny monolingüe, en català, i deixar la opció de convertir-lo en multilingüe per a més endavant. Amb aquesta idea, s'ha considerat preferible no utilitzar aquestes taules però tampoc esborrar-les, ja que tenen les relacions entre elles creades i això facilitarà la tasca futura d'afegir altres idiomes.

7. Models i controladors

La part de lògica en PHP està separada de la part visual en HTML i CSS. Dins d'aquesta part lògica destaquen els models i els controladors, i a més, també són importants per al disseny del lloc web les rutes.

Per exemple, quan s'escriu en el navegador la direcció d'un article del web, la ruta envia la informació, amb la dada de l'*slug* al controlador, aquest envia el requeriment de la informació a través del model corresponent a la taula dels articles cap a la base de dades, i des de la base de dades es retorna les dades completes de l'entrada d'aquest article en particular. El controlador les consulta i les envia cap a l'arxiu de la vista genèrica *post.blade.php*, on s'extraurà la informació directament per publicar-la.

Tant models com controladors tenen alguns elements comuns. Els arxius segueixen una estructura semblant:

1. Declaració de quin és el *namespace* (nom de l'espai). Les funcions i les classes de cada arxiu estan sota el paraigües d'aquest nom, de manera que puguin existir funcions amb el mateix nom sempre i quan estiguin en diferents *namespace*. Això facilita la comprensió, ja que els noms de les funcions poden ser molt més descriptius. El *namespace* també s'utilitza per ubicar les funcions i classes.
2. Declaració dels arxius utilitzats com a referència, amb la localització perquè *Laravel* els pugui trobar quan siguin necessaris. D'aquesta manera es poden utilitzar funcions que estan en altres arxius.
3. Obertura i nom de la classe dins de la qual es crearan les funcions.
4. Extensió d'una altra classe. La majoria de models estenen la classe *Model* i la majoria de controladors, la *Controller*. En tots dos casos fan referència als arxius base creats per *Laravel*. En alguns casos es pot estendre un model o controlador anterior amb el mateix nom de classe. Això s'utilitza per ampliar el contingut d'aquest model o controlador.

7.1. Models

Els models són la part del codi dedicada a la connexió amb la base de dades. Utilitzen *Eloquent* [16], que és el programari que fa servir *Laravel* amb aquest objectiu. Quan es crea una nova taula a la base de dades en molts casos s'hi ha de vincular un model nou, que servirà per fer referència a aquesta taula en particular. Les excepcions són les taules a les que no cal fer referència directament, que són les taules de suport. En aquest projecte, les

úniques taules de suport sense model propi són les utilitzades per crear relacions entre taules, que només contenen els números identificatius de les taules que s'hi vinculen.

Creant un arxiu per a una taula en particular i estenent la classe *Model* el que s'aconsegueix és un model amb totes les funcions de l'arxiu original, però sota el *namespace* d'aquesta taula. Per a alguns models generats amb això n'hi ha prou, perquè ja permet realitzar les accions més habituals relacionades amb taules.

En aquest projecte això s'ha fet amb el model de la taula sales, que s'ha anomenat **Sales**. És necessari fer-ne alguna crida des dels controladors, com es veurà més endavant, però no necessita cap funció addicional.

Un cas semblant s'ha donat amb el model **Publics**. En aquest cas, només s'ha afegit una línia, que és per protegir el nom de la taula. *Laravel* per defecte assumeix que els noms de les taules acaben en s, però en aquest cas es volia que la taula es digués *public*. Protegint el nom de la taula es resol aquest problema. També s'ha fet el mateix amb el model **Directori** per a la taula *directori*.

D'altra banda, per a les taules creades per *Voyager* no cal generar nous models, perquè *Voyager* també inclou els seus models. Per utilitzar-los només cal referenciar-los adequadament, com per exemple s'ha fet amb *Post* de la taula on estan els articles.

El model **User** és una mica diferent, perquè també es genera quan s'instal·la *Voyager*, però en dues parts. Un està a la carpeta on estan tota la resta de models de *Voyager* i l'altra amb els de l'aplicació. Aquest últim estén a l'anterior, de manera que formen una sola classe en dos *namespace* diferents. A banda de les existents, que serveixen per afegir atributs a les columnes de la taula de persones usuàries, s'hi ha afegit una funció per a enviar les notificacions per als canvis de contrasenya amb el text personalitzat. L'única cosa que fa aquesta funció és redirigir al notificador creat amb aquest objectiu.

El model que s'ha desenvolupat més és **Activitat**. A la declaració dels arxius de referència s'hi ha agregat *Carbon* [17][18], que inclou mètodes que faciliten el tractament de les dates. Dins de la classe, s'ha indicat que les columnes referents a les dates d'inici i finalització i la data límit per a inscripcions tenen el format de data. Això permetrà utilitzar els mètodes de *Carbon* en els controladors en què s'utilitzin. De manera semblant, també s'ha indicat que les columnes d'hora d'inici i hora de finalització tenen el format d'hora.

S'hi ha afegit la funció *save*, amb la que es dona un valor per defecte al camp de direcció, utilitzant el valor escrit a l'apartat de configuració del panell d'administració. Això facilitarà que si en algun moment cal canvia la direcció principal es pugui fer fàcilment sense tocar el codi del lloc web.

A continuació hi ha les funcions de relació. Totes dues indiquen una relació entre aquesta taula i la taula de persones usuàries, una és per a les persones assistents i l'altra per a les representants. D'aquesta manera, es podran utilitzar per vincular i desvincular les taules amb els mètodes de *Eloquent* per a relacions. Si només es fes a través del panell d'administració no caldria fer-ho, perquè ja s'ha creat la relació amb *Voyager*, però s'utilitzaran també en altres punts del lloc web.

7.2. Notificador

Les notificacions [19] s'utilitzen per enviar correus electrònics. Quan es genera el sistema de registre i d'inici de sessió que *Laravel* inclou per defecte, s'hi inclou la funcionalitat d'enviar un correu electrònic que permet recuperar la contrasenya en cas d'oblit. Amb aquesta opció per defecte, el contingut del correu electrònic està predefinit. Per a poder-lo sobreescrivre i personalitzar el text cal seguir diferents passos.

El primer és crear un arxiu de notificacions. Les funcions per defecte estan a la part del codi base de *Laravel* que convé no modificar. El nou arxiu s'utilitzarà per sobreescrivre l'anterior sense tocar el codi base. Per crear-lo i que segueixi fent les mateixes funcions s'ha copiat tot el codi de l'anterior i només s'ha canviat la part referent al contingut del missatge. Aquest arxiu es guarda a la carpeta creada a aquest efecte dins de la carpeta *app*.

També cal assegurar-se que a l'arxiu *.env* hi ha el nom de l'aplicació correcte, ja que s'utilitza en el text del correus electrònics.

A més, també s'ha de crear un arxiu i una carpeta dins de les vistes on es mostra el contingut dels correus per defecte (és a dir, en cas que no se sobreescrigui). Des d'aquí es pot canviar el text referent als enllaços compartits.

7.3. Rutes

Les rutes [20] serveixen per relacionar una direcció escrita en el navegador amb l'arxiu de la vista corresponent. També s'utilitzen com a pont entre vistes i controladors. A algunes rutes a més a més se'ls assigna un nom, cosa que facilita referenciar-les.

Totes les rutes han d'estar indicades a l'arxiu *web.php* que es troba a la carpeta de rutes. En alguns casos en els que cal afegir més contingut, s'ha escrit la funció de la ruta directament en aquest arxiu. Per a casos més complexos o en els que es vulgui agrupar diverses rutes

d'una temàtica, s'utilitzen els controladors. En aquest cas, per a rutes úniques, a l'arxiu *web.php* s'hi indica la direcció *url*, el controlador i la funció del controlador que s'utilitza en aquesta ruta.

Els paquets *Auth* i *Voyager* generen les seves pròpies rutes. Aquest últim permet personalitzar-les i així s'ha fet amb la seva ruta per a tancar sessió. Per defecte retornava a la pàgina d'inici de sessió per al panell d'administració i es va considerar que era poc pràctic. S'ha sobreescrit dins de *web.php* per utilitzar la funció *logout* del controlador del paquet *Auth*, amb el que es retorna a la pàgina principal.

En alguns casos és necessari enviar algun paràmetre a través d'una ruta, per compartir informació entre una vista i un controlador. Per fer-ho cal que estigui indicat a la direcció *url*, tant si és un sol paràmetre com si en són més. Per exemple, en el cas de la ruta per mostrar un article, s'afegeix l'*slug* com a variable i això permet al controlador buscar quin és l'article que ha d'enviar a la vista corresponent.

Es poden utilitzar diversos verbs per a les rutes i cada un comporta diferents accions. En aquest cas s'han utilitzat els verbs *get*, *post*, *put* i *patch*. El primer, que és el més comú, és per a les rutes en les que es mostren resultats, com ara quan es mostra una activitat en concret. El *post*, per a la majoria de casos en els que s'envia informació d'un formulari des d'una vista cap a un controlador. Amb les excepcions dels formularis que fan servir *put*, que s'utilitza quan es vol modificar una entrada sencera d'una taula de la base de dades, o *patch*, per als casos en que només cal canviar algun dels camps o una relació.

7.4. Polítiques d'autorització

En alguns casos cal autoritzar o restringir l'accés a algunes pàgines a algunes persones usuàries. Això es pot fer de diverses maneres. D'una banda, els *middleware* s'encarreguen de comprovar si una persona pot accedir a les pàgines i redirigir-la en funció del resultat. Tots els *middleware* que hi ha a l'aplicació s'han generat a través de *Laravel* o els paquets *Auth* i *Voyager*.

D'altra banda i de manera complementària, s'han creat arxius de polítiques [21], que és on s'han escrit les funcions que defineixen les condicions per les que una determinada persona pot accedir a alguns apartats del lloc web. De manera semblant al que es fa amb rutes i controladors, primer cal declarar les polítiques a l'arxiu *AuthServiceProvider.php*.

S'han creat dos arxius de polítiques, per a les activitats i per al directori. Per a la resta no cal crear-les perquè les restriccions necessàries les gestionen *Auth* i *Voyager*. Tant per a les

polítiques d'activitats com per les del directori s'ha seguit la mateixa estructura, s'hi han generat dues funcions a l'interior de les respectives classes, una per definir qui pot editar i una per qui pot esborrar. Les dues funcions permetran aquestes accions només quan qui les realitzi sigui una de les persones responsables de l'activitat o de l'entrada del directori, o bé quan ho faci una persona amb el rol d'administració.

Les polítiques per autoritzar accions queden vinculades als respectius models. Per utilitzar-les, es poden cridar les funcions des de les rutes, els controladors o les vistes.

7.5. Controladors

Els controladors [22] són la part del codi que connecta les vistes amb els models. S'utilitzen per reduir al mínim a part de lògica que hi ha d'haver a les vistes i perquè una mateixa funció es pugui utilitzar en diverses vistes. A diferència dels models, no estan vinculats a una sola taula de la base de dades sinó que n'utilitzen varies, sempre a través del model corresponent.

Tant *Voyager* com *Auth* tenen els seus propis controladors que no s'explicaran en aquest apartat. Com en el cas dels models, hi ha un controlador bàsic que la resta de controladors utilitzen fent-ne una extensió. A banda d'aquests, s'han creat cinc controladors, un per als articles, un per a les activitats, un per al directori, un per al perfil i un per a la pàgina principal. En els següents subapartats s'explicaran amb més detall.

7.5.1. Controlador per als articles

Els articles només es podran crear i modificar amb el panell d'administració, és a dir que aquestes funcionalitats les gestionen els controladors de *Voyager*. En el controlador per als articles (*PostController.php*), hi ha dues funcions: per a mostrar una llista d'articles i per mostrar un sol article.

La primera és la funció *index*. Primer demana a la base de dades que busqui els elements de la taula d'articles que compleixin els requisits: tenir l'estat de *publicat* i que el seu camp *deleted_at* sigui nul. A continuació els ordena segons la data de creació (*created_at*) en ordre descendent. I de tots aquests elements, que n'agafi deu. Finalment, retorna la vista corresponent, enviant-li la informació resultant.

Per a mostrar un article es fa servir la funció *show*, que demana un paràmetre, que en aquest cas es correspondrà amb l'*slug*. Amb aquesta dada, es fa una petició a la base de

dades perquè trobi l'article amb aquest *slug*, que estigui en estat publicat i que tingui el camp *deleted_at* amb valor nul. Amb el mètode *firstOrFail* es demana a la base de dades que retorni el valor trobat i que, si no existeix cap element que compleixi aquest requisits, retorni error. En aquest últim cas, es redirigeix a la pàgina d'error *pàgina no trobada*.

En canvi, si retorna un article, a continuació es comprova si té un valor d'autoria assignada. L'autoria és una relació amb la taula de persones usuàries que es suporta amb una taula complementària. Per com es guarda la informació a les taules quan es crea un nou article, no hi hauria d'haver cap article sense autoria, però si es donés el cas (per exemple, perquè s'ha omplert la taula directament i no a través del panell d'administració) i es busqués directament qui és la persona usuària vinculada per l'autoria, es produiria un error. Per evitar-lo, es crea un variable per a l'autoria i en aquest cas se li assigna valor "*La Trobada*". Si l'autoria ja existia, se li assigna a aquesta variable el nom de la persona usuària vinculada.

Finalment, es retorna la vista per a un article, a la que s'envia l'article i la variable de l'autoria.

7.5.2. Controlador per a les activitats

En el controlador per a les activitats també hi ha les funcions *index* i *show*, amb algunes diferències respecte les funcions amb el mateix nom del controlador per als articles. A més, s'han escrit també les funcions *create*, *store*, *edit*, *update*, *destroy*, *atAssistent* i *detAssistent*.

Per a *index*, es fa una petició a la base de dades per trobar les activitats que tinguin una data d'inici igual o superior a la data actual (la del dia en que es realitza la consulta), que tingui com a estat *Publicada* i tingui el camp *deleted_at* nul. Després s'ordena primer per la data d'inici i després per la hora d'inici, de manera que les més recents estiguin abans. Existeix la possibilitat que aquesta cerca retorni un conjunt buit, si fos així, s'enviaria a una pàgina on es digués que actualment no hi ha activitats programades. En cas contrari, es retorna la vista on es mostrarà el llistat d'activitats, enviant-li la informació de les activitats.

Per a *show*, cal informació també d'altres taules. Primer es demana a la base de dades l'activitat que tingui l'*slug* indicat, que pugui ser publicada i que no estigui esborrada. Si no en troba cap, retorna l'error de *pàgina no trobada* i retorna a la vista corresponent.

En cas contrari a continuació es crea una variable per guardar-hi les persones assistents i una altra per les persones responsables. Es troben els elements de la taula de persones usuàries en tots dos casos gràcies a les funcions definides en el model *Activitat* per vincular

les taules. Després es comprova si hi ha una sala assignada a aquesta activitat i si no és així, a la variable *sala* se li assigna el valor “Espai pendent de decidir”.

Totes aquestes dades s'enviaran a la vista corresponent conjuntament amb l'activitat. Se selecciona la vista depenent de si l'activitat és per a un o per més dies i segons el tipus d'activitat que sigui.

La funció *create* simplement retorna la vista on hi ha el formulari per crear una nova activitat. I un cop s'envia la informació des d'aquesta, la gestiona la funció *store*. En aquesta, s'agafen les dades de cada camp del formulari i es guarden com a camps de la taula d'activitats generant una nova entrada. L'estat de la nova activitat serà *Pendent* (de revisió), perquè caldrà que una de les persones encarregades del manteniment i la gestió del lloc web en doni la seva aprovació. Després es redirigeix a la ruta d'editar l'activitat.

La funció *edit* comprova que l'activitat que es vol editar té un valor nul per a *deleted_at* i retorna la vista del formulari d'edició. Per a fer els canvis, s'utilitzarà la funció *update*, amb la qual es canviaran tots els camps de l'activitat a la taula amb els del formulari. A continuació es fa una redirecció a la mateixa pàgina d'edició. En cas que el que es vulgui sigui esborrar-la, es faria servir la funció *destroy*. El que fa primer és eliminar totes les relacions d'aquesta activitat amb altres taules i després hi afegeix el camp *deleted_at*. Fer-ne un esborrat real (és a dir, eliminar l'entrada de la taula d'activitats), serà una tasca de manteniment.

Les dues últimes funcions, *atAssistent* i *detAssistent*, permeten respectivament generar un vincle amb la taula de persones usuàries i eliminar un vincle existent. S'utilitzen perquè una persona pugui apuntar-se o desapuntar-se d'una activitat.

7.5.3. Controlador per al directori

Les funcions del controlador de directori són molt semblants a les del controlador de les activitats. En aquest cas són *index*, *create*, *store*, *edit*, *update* i *destroy*. No s'ha inclòs una funció *show* perquè no es mostrarà cap entrada del directori en una pàgina independent.

Amb *index* es generarà una llista d'entrades del directori, entre les que estiguin publicades i no s'hagin eliminat, i s'ordenarà primer segons el tipus (persona o entitat) i després pel nom de l'entrada. Si la llista està buida, es retorna una vista amb un missatge informant de que el directori està buit. En cas contrari, s'envia la llista a la vista principal del directori.

Les funcions *create* i *store* seran les mateixes que per a les activitats. La primera retorna la vista del formulari i la segona emmagatzema la informació creant una nova entrada a la taula. També amb *edit* i *update*, es comprova que l'entrada no ha estat esborrada (verificant

el camp *deleted_at*), es retorna la vista del formulari d'edició i posteriorment es validen i es guarden les noves dades a l'element corresponent de la taula del directori. Si es vol esborrar, amb la funció *destroy* primer es desvincula aquesta entrada del directori respecte altres taules i després s'afegeix la marca *deleted_at*.

7.5.4. Controlador per al perfil

La pàgina del perfil d'una persona usuària té la particularitat que està formada per blocs relacionats amb diferents taules. Per tant, en el seu controlador cal treballar amb aquestes taules, és a dir, cal referenciar els models de la taula de persones usuàries, la d'activitats i la del directori. Hi ha les funcions *show*, *edit*, *update* i *destroy* i com a funció complementària, *validator*.

Per a la primera, la funció demana el paràmetre *user* (persona usuària). A partir d'aquí, troba les activitats a les que ha d'assistir i a les que és responsable aquesta persona. També en busca l'entrada del directori. Això es fa gràcies a les funcions de relació creades en el model *User*. A continuació envia tota aquesta informació cap a la vista del perfil.

Amb *edit*, es mostra el formulari per canviar les dades personals. Per validar la informació provinent del formulari, s'utilitza *validator*, que es crida dins d'*update*. Dins d'aquesta funció, després de validar es comprova la quantitat de vegades que s'ha intentat enviar el formulari, si en són masses, no permet tornar-ho a intentar durant un minut. A continuació es comprova si la contrasenya concorda amb la de la persona que té iniciada la sessió. Si és així, es guarden els camps del formulari i es retorna la vista del perfil. En cas contrari, s'incrementa el nombre d'intents i es retorna al formulari sense guardar cap canvi.

Per a la funció *destroy* el procediment és semblant, també es validen les dades, es limiten els intents seguits i es comprova la contrasenya. Si s'aconsegueix tot això, primer s'eliminen les relacions amb altres taules que tingués aquest element *user* i després s'elimina de la taula. No es guarda cap informació personal per mantenir el dret a la pròpia informació de les persones usuàries.

7.5.5. Controlador per a la pàgina principal

Per a la pàgina principal només cal una funció que en gestioni la vista. Com el controlador tindrà una sola funció, no cal donar un nom a la funció.

Per una banda, fa una cerca a la taula d'activitats semblant a la realitzada per a crear el llistat de la pàgina d'activitats. En aquest cas, se'n seleccionen, com a molt, cinc. Per una altra banda, fa una altra cerca, en aquest cas com la realitzada per al llistat de la pàgina d'articles, però triant-ne, com a molt, cinc també.

A més es fa una nova cerca per buscar quin ha de ser l'element destacat de la pàgina principal. Primer, entre els elements de la taula d'activitats, seleccionant les que estan publicades, no han estat esborrades, comencen en la data actual o en una posterior i estan marcades com a destacades. Entre les trobades, s'ordenen per dia d'inici i es tria la primera. En cas de que no es trobi cap activitat que compleixi els requisits, es busca a la taula d'articles els publicats, no esborrats i marcats com a destacats. Aquests s'ordenen per la data de creació i es tria el més recent. No es fan més cerques, en cas que no hi hagi element destacat es gestionarà a la vista. Tota la informació trobada s'envia a la vista *index*.

8. Vistes

Les vistes són la part del codi que fa referència al que apareix en pantalla més directament i en general s'escriuen en HTML i poden tenir alguns elements en PHP i en javascript. *Laravel* utilitza *Blade* com a gestor de les vistes i d'aquesta manera simplifica tant la creació com la lectura del codi, dividint-lo en arxius més curts i més clars.

Les vistes s'organitzen per carpetes temàtiques. Per exemple, les vistes relacionades amb el sistema de registre i d'inici de sessió estan a la carpeta *Auth*. Algunes carpetes tenen elements comuns per a totes les vistes, com la carpeta de plantilles.

El panell d'administració té les seves pròpies vistes, generades per *Voyager*, igual que té els seus propis controladors, rutes i models. No es descriuran aquestes vistes i les pàgines que generen s'han explicat a l'apartat [5.3](#).

8.1. Plantilles i parcials

Per reduir les feines repetitives, s'utilitza una plantilla principal, on hi ha l'estructura de la pàgina, la capçalera i el peu. Aquesta plantilla es referència a totes les vistes que es creen posteriorment, de manera que s'assegura una coherència estructural i visual en totes elles. La barra de menú de navegació s'ha afegit a la plantilla des d'un arxiu separat, utilitzant la directiva *include*.

Aquesta directiva permet injectar fragments de codi des d'altres arxius i això s'utilitza per simplificar cada arxiu i per compartir les parts de codi comú en diferents pàgines fent referència a un sol arxiu. Aquests fragments de codi es guarden a les carpetes de *parcials* dins de les carpetes on s'utilitzen.

La plantilla de la barra de navegació es construeix amb el suport de Bootstrap. La majoria d'elements queden a l'esquerra de la barra, mentre que a la dreta hi ha l'element de perfil, que canvia en funció de si la persona usuària ha iniciat sessió i dels permisos que tingui. Per fer-ho, s'han utilitzat les estructures de control de *Auth* [23] i de *Voyager* [24], que validen l'autenticació i les accions que es poden fer.

8.2. Errors de visualització

Els errors relacionats amb les vistes més comuns són el 403 i el 404, que fan referència respectivament a intentar accedir a una pàgina que requereix uns permisos que no es tenen i a intentar accedir a una pàgina inexistent. Per a cada una s'ha creat una pàgina on es mostra un missatge d'error descriptiu. En cas de produir-se aquests errors, es fa una redirecció a aquestes pàgines, de manera que la persona usuària té la informació per decidir com resoldre el problema.

També es poden produir errors més localitzats a l'interior d'alguna pàgina, principalment en formularis. En aquests casos s'emeten missatges en els punts on s'han produït els errors.

8.3. Pàgina principal

La pàgina principal visualment té un element principal destacat, sota del destacat dues columnes on hi apareixen activitats i notícies, i una columna a la dreta amb un petit mapa amb l'adreça del local i diversos missatges per convidar a participar en l'espai.

El mapa s'enllaça des de *Google Maps* [25], utilitzant el codi de la pròpia aplicació, adaptant-ne les mides en el codi per ajudar-les a l'ample de la columna. El text que hi ha sota del mapa és fix i està escrit directament en el codi.

L'element destacat serà una activitat o una notícia triada. La imatge serà més gran que per a la resta d'elements i estarà guardada separada de les imatges internes de cada activitat o article. S'enllaça directament des d'una de les carpetes d'imatges del panell d'administració, utilitzant les rutes de *Voyager*. Amb la informació enviada pel controlador, es genera un enllaç a l'element corresponent a la imatge.

Les activitats i notícies que es veuran a sota es col·loquen de manera semblant a com es generen les seves vistes de llista d'elements. Estan formades per blocs on hi ha una imatge, un títol i un breu text descriptiu. El filtratge per triar quines es trien es fa a través del controlador.

8.4. Vistes per al perfil

A la informació del perfil personal només hi pot accedir una persona registrada i ha de tenir iniciada la sessió. Per veure aquesta pàgina, en el menú principal hi apareixerà l'accés, junt

amb altres accions possibles per gestionar el perfil sense necessitat d'entrar en aquesta pàgina.

La vista d'aquesta pàgina està dividida en diferents blocs. En el primer apareixen les dades personals de registre: el nom utilitzat, l'adreça electrònica, la contrasenya en caràcters ocults, la imatge d'avatar i el rol assignat. A més hi ha un botó que redirigeix a la pàgina d'edició de les dades personals.

A continuació hi ha el bloc de les activitats. Primer estan les activitats de les que és responsable la persona que veu el perfil i després les activitats en les que s'ha inscrit per assistir-hi. Per a les primeres hi ha un enllaç per veure la pàgina corresponent a l'activitat i un altre per editar-la, mentre que per a les últimes només estan els enllaços per visualitzar-les.

Finalment està l'apartat del directori, on es pot veure la informació que s'hi ha publicat i hi ha un botó que redirigeix a la pàgina per editar-la.

Si més endavant s'afegissin altres funcionalitats per al perfil, es veurien en blocs a continuació d'aquests.

El botó d'edició de la informació personal redirigeix al controlador corresponent i aquest redirigeix alhora a la pàgina d'edició. En aquesta última es poden canviar tots els camps anomenats excepte el de la contrasenya. Els camps del nom, l'adreça electrònica i la contrasenya (per verificar-la) estan marcats com a necessaris, de manera que no es pugui enviar la informació si estan buits. També s'indica que el camp de l'adreça electrònica ha de tenir aquest format, de manera que el propi navegador en faci una comprovació abans d'enviar-se. Aquestes comprovacions prèvies eviten enviar informació innecessària, reduint el temps d'espera i els recursos empleats. Quan la informació s'ha passat la comprovació del navegador, s'envia al controlador corresponent perquè la gestioni.

8.5. Vistes de registre i inici de sessió

Les vistes per al registre i l'inici de sessió es creen automàticament amb el paquet *Auth* de *Laravel*. En total són cinc vistes:

- Formulari de registre
- Formulari d'inici de sessió
- Formulari per demanar un correu per poder restablir la contrasenya

- Formulari per restablir la contrasenya
- Confirmació de l'inici de sessió

En tots els formularis hi ha verificació tant en els mateixos camps prèvia a l'enviament com en el servidor. Els missatges d'error que poden aparèixer estan guardats en un arxiu apart, de manera que se simplifica la tasca de traduir-los i permetria tenir-los en diferents idiomes, si es volgués. *Laravel* genera l'arxiu en anglès i s'han traduït al català, deixant la versió en anglès com a secundària (només s'utilitzaria en cas de no trobar els arxius en català).

A més del canvi d'idioma, s'han fet altres canvis, com afegir una casella per al consentiment, alguns canvis menors que afecten a l'estètica i vincular aquestes vistes amb la plantilla general.

8.6. Vistes per a les activitats

Per a les activitats s'han creat una vista per al llistat d'activitats, diverses per a mostrar una activitat concreta, una per al cas que no hi hagi activitats programades per mostrar al llistat, dues per als formularis per enviar propostes d'activitats i per editar les activitats existents.

Dins de la vista del llistat, s'ha creat un bucle que repassa totes les activitats enviades des del controlador. Per a cada cicle, s'afegeix un element a la llista, amb el títol, el tipus d'activitat, el dia i hora d'inici, el resum i la imatge en petit (si n'hi ha). A més s'hi afegeix un enllaç per veure l'activitat amb més detall.

Hi ha diverses vistes per mostrar una activitat, una és per a les activitats d'un sol dia (siguin del tipus que siguin), una per a les jornades i una per als cursos. A més, en totes elles s'hi inclou una vista parcial. Les vistes es diferencien entre si per quina informació mostren i com la mostren. A les activitats puntuals hi ha el títol, la imatge, quin tipus d'activitat és, on es farà, el dia i l'hora d'inici, la data de finalització i la descripció de l'activitat. A continuació és on s'inclou la vista parcial. També hi ha els botons per a editar o esborrar l'activitat, que només es mostren a les persones responsables de l'activitat.

Per a la vista de jornades, en canvi, per la localització no es concreta la sala o l'espai, sinó només la direcció, i a més a la data de finalització hi afegeix l'hora. Per a la del curs, s'hi diu les dates d'inici i de finalització, sense posar hores, i després s'hi inclou l'horari que tindrà el curs. En tots els casos, cal fer un canvi en el format de les dates i les hores per adaptar-les al format català, ja que a la base de dades es guarden amb un format anglosaxó.

A la vista parcial en primer lloc hi ha una comptabilització del nombre de persones que s'han inscrit en aquesta activitat fins al moment. El text varia si és una persona o si són més d'una. Si la persona no està registrada, es mostra un missatge convidant a registrar-se o iniciar sessió per a apuntar-se a l'activitat. En cas contrari, es comprova si és una de les persones responsables de l'activitat. Si ho és, se li mostrarà un llistat dels noms de les persones inscrites. A continuació hi ha el botó per apuntar-se o desapuntar-se. Per triar quin surt, amb un bucle es repassen les persones assistents i es comprova si la persona que està activa a la pàgina és una d'elles. Després del bucle si no hi ha hagut coincidència, se selecciona l'opció de mostrar el botó per desapuntar-se.

La vista per enviar una nova activitat consisteix principalment en un formulari. Enviarà la informació amb el mètode *post* a la ruta per guardar una nova entrada a la taula. Dins del formulari hi ha els camps necessaris per fer una proposta d'activitat. El títol i el tipus d'activitat s'han d'omplir per a poder enviar les dades.

A la vista d'edició hi ha un formulari amb els mateixos camps, però en aquest cas contenen la informació que ja hi havia a la taula, de tal manera que si es guardés directament, no s'hi produirien canvis. Això facilita la tasca d'actualitzar la informació. A més, el mètode d'enviament no és *post* sinó *put*.

8.7. Vistes per als articles

Els articles només es podran escriure des del panell d'administració, així que només s'han creat dues vistes, una per al llistat d'articles i l'altra per a mostrar-ne un en particular.

El llistat d'articles és semblant al d'activitats. Es llisten a partir d'un bucle, amb el títol, el resum i la imatge (reduïda), i després hi ha un enllaç amb la ruta de l'article.

A la vista on es mostra un de sol, hi ha directament el títol, la imatge, el cos de l'article i l'autoria, amb la informació rebuda des del controlador.

8.8. Vistes per al directori

S'han creat tres vistes per al directori, l'índex, el formulari de creació d'una nova entrada i el formulari d'edició. No es veuran les entrades del directori de manera separada.

L'índex és un llistat de totes les entrades del directori rebudes des del controlador. Se separa entre persones i entitats. Per a cada entrada hi ha el nom, la descripció, una imatge i les dades de contacte. Tant el nom com les dades de contacte poden ser diferents als de la persona usuària que l'ha creat. A més del llistat, les persones no registrades veuran un text convidant a registrar-se o iniciar sessió per poder afegir una entrada. A les que tinguin la sessió iniciada i no tinguin cap entrada, els apareixerà un text convidant a afegir una entrada i un botó per fer-ho. I les que ja en tinguin una veuran el botó d'edició.

Cada persona, com a molt, podrà crear una entrada del directori. Això ho podrà fer amb el formulari de noves entrades a la vista corresponent. Hi ha els mateixos camps que després es mostraran i la informació s'envia amb el mètode *post*. No es publicarà directament, perquè per defecte l'entrada es guarda amb l'estat com a *inactiu*, fins que la revisi una de les persones encarregades del manteniment del web.

A la vista d'edició, hi ha un formulari semblant però amb l'entrada que es vol modificar en els camps corresponents. Aquest envia la informació amb el mètode *put*.

8.9. Vistes de les pàgines de presentació

La vista per a la pàgina del **projecte** està formada únicament per una imatge i el text de descripció del projecte. La de l'**equip** del projecte, conté un text introductori i a continuació una llista amb els noms de les persones que formen el col·lectiu, amb una imatge i un text breu descriptiu.

La vista sobre la **localització** conté una primera part amb una llista de dos elements, on hi ha l'adreça i la informació del transport públic proper a l'espai físic. La segona part consta del mapa de la localització, que és de *Google Maps* i s'inclou amb el codi que ofereix per a enllaçar-lo. S'introdueix dins d'un element HTML que permet un comportament *responsive*, és a dir, que s'adaptarà en funció de la mida de la pantalla.

La pàgina de **contacte** està dividida en dues parts, la primera amb un llistat de dades i la segona amb un formulari. Dins de la vista, per a la primera part s'ha generat una llista d'elements. Hi ha l'adreça postal de l'espai físic, l'horari d'obertura, el número de telèfon amb el seu horari d'atenció, l'adreça electrònica i les xarxes socials.

La segona part té el formulari de contacte que utilitza *Formcarry* per a l'enviament de correus electrònics. Tots els camps estan marcats com a necessaris. Es demana el nom, una adreça electrònica, el cos del missatge i que s'acceptin dels condicions d'ús. Dins del

text de les condicions d'ús s'inclouen enllaços tant a la pàgina d'informació legal d'aquest lloc web, com a les de *Formcarry*.

8.10. Vistes de les pàgines de serveis

Per crear la pàgina de les **sales i espais**, la informació de la base de dades s'envia a la vista directament a través de la ruta, és a dir, no s'ha creat un controlador per realitzar aquesta funció. En el text introductori, s'enllaça a la ruta de la pàgina de contacte com a mètode per a realitzar la reserva de sales. A continuació hi ha el llistat de les sales, que es genera amb un bucle. En cada element hi ha el nom, la descripció, la capacitat i una imatge de la sala.

La vista de la pàgina de **cotreball**, està formada principalment per una llista on hi ha les diferents tarifes i la informació de què inclou i en quines condicions cada tarifa. Després del llistat, hi ha un text demanant que en cas d'interès es posin en contacte i s'hi inclou un enllaç a la pàgina de contacte.

A la vista de la pàgina d'**exposicions** hi ha un text genèric introductori i a continuació algunes imatges d'exemple. En el text introductori s'hi enllaça la pàgina de contacte demanant a les persones interessades en exposar que es posin en contacte. Les imatges de mostra s'enllacen a través del panell d'administració, amb els mètodes de *Voyager*, de manera que es puguin reemplaçar des d'allà.

8.11. Vista per a les condicions d'ús

En aquesta vista s'inclou el text legal sobre les condicions d'ús de la pàgina, el tractament de dades i quines són les normatives i lleis de referència que s'estan aplicant. L'únic contingut de la vista és aquest text dividit per temes.

A aquesta pàgina s'hi pot arribar tant des del peu de pàgina (accessible des de qualsevol punt del lloc web), com des dels formularis en els que s'envien dades personals.

9. Seguretat i tractament de dades

La seguretat és una part important de qualsevol lloc web. Cal tenir en compte tant els possibles atacs maliciosos que cal prevenir o frenar, com la normativa existent. No es pot aconseguir una certesa total pel que fa a la seguretat de cap lloc web, però es poden reduir notablement els riscos.

9.1. Normativa

Tal com s'ha indicat a l'apartat [3.2](#), cal complir la normativa de protecció de dades, concretament el Reglament general de protecció de dades (RGPD) i la Llei orgànica de protecció de dades (LOPD).

Amb aquest objectiu s'han seguit els següents passos:

- S'ha creat una pàgina explicant quines són les lleis aplicades, quin és el protocol de tractament de dades, les condicions d'ús del lloc web, les condicions per a registrar-se i què cal fer per modificar o eliminar les dades personals enviades, revocant-ne el consentiment d'ús.
- Les dades guardades per a la creació del perfil i altres dades personals que s'hagin guardat, es poden eliminar de dues maneres: a través del formulari del propi lloc web o bé enviant un correu electrònic notificant la voluntat de donar-se de baixa. També es poden modificar en qualsevol moment a través del formulari del web.
- En el formulari de registre i en qualsevol altre formulari que impliqui enviar informació personal s'hi ha afegit una casella per donar el consentiment per a l'ús d'aquestes dades. Junt a aquesta casella hi ha un breu text explicant què es farà amb aquestes dades i un enllaç a la pàgina on s'explica la normativa més detalladament.
- Per emmagatzemar la informació s'han seguit diversos protocols de seguretat, com es veurà en els següents apartats.
- Les dades personals només es guarden en el servidor del lloc web, no hi ha altres empreses involucrades en el tractament de dades. Només en el cas del formulari de contacte, la informació enviada és guardada a *FormCarry* i així s'indica a la mateixa pàgina del formulari, dins del text per demanar el consentiment.
- El registre a través del formulari del web només es permet a persones majors de setze anys. Per a menors d'aquesta edat caldria seguir un procediment alternatiu que permetés assegurar que existeix també un consentiment per part de pares o representants legals.

9.2. Contrasenya i encriptació

Les contrasenyes són un element especialment sensible. No només per a la seguretat de la informació personal en el lloc web, també perquè és una pràctica habitual (tot i que poc recomanada) utilitzar la mateixa contrasenya en diversos web. Per aquesta raó és necessari que aquesta dada estigui especialment protegida.

Quan s'envia la informació, la contrasenya s'encripta per guardar-la a la base de dades. D'aquesta manera, encara que s'accedeixi a la base de dades no es pot veure la contrasenya, sinó només un codi corresponent a la seva encriptació. S'han utilitzat les funcions d'encriptació que hi ha per defecte a *Laravel*, que estan explicades a la seva documentació [10]. S'ha considerat que utilitzar aquestes funcions és una opció més segura que qualsevol intent de generar funcions pròpies.

S'ha generat també un formulari per recuperar la contrasenya, també amb el sistema per defecte de *Laravel*, *Auth*. Per a fer-ho, es crea un *token* a la base de dades i s'envia també per correu electrònic. La validesa d'aquest *token* és com a màxim d'una hora des de la seva creació. Per recuperar la contrasenya es comprova que els valors de *token* concorden i a continuació es demana a la persona usuària que guardi una nova contrasenya, que s'encriptarà de la mateixa manera. Aquest procediment permet tornar a entrar si es té accés a l'adreça de correu electrònic però sense revelar en cap cas quina era la contrasenya original.

9.3. Protocol (HTTPS)

Encara que en el servidor es codifiqui la informació, s'ha de tenir en compte també el trajecte de les dades des del dispositiu client fins al servidor. Molts llocs web segueixen un protocol d'enviament de dades HTTP (*Hypertext Transfer Protocol*, és a dir, protocol de transferència d'hipertext). Aquest protocol té l'inconvenient que no xifra la informació quan s'envia i per tant si s'intercepta durant l'enviament res evita que l'atacant pugui utilitzar les dades directament [26].

Per aquesta raó quan s'envia informació cal utilitzar protocols més segurs, com HTTPS (*Hypertext Transport Protocol Secure*, protocol segur de transferència d'hipertext). A més, no és suficient que el formulari estigui en HTTPS, cal que ho estigui la pàgina sencera on està el formulari i si és possible tot el lloc web [27]. Això depèn de la configuració del servidor i per tant com a mesura de seguretat s'ha decidit allotjar el lloc web en un servidor configurat per funcionar amb HTTPS.

9.4. Mesures de prevenció d'atacs

Algunes mesures per prevenir atacs al lloc web són:

- Limitar la quantitat d'intents seguits per iniciar sessió. Per defecte, si s'intenta iniciar sessió sense èxit cinc vegades seguides, ha de passar un minut abans que es pugui realitzar un nou intent. Això es fa amb la configuració per defecte de *Laravel*, amb el mètode *ThrottlesLogins* [28].
- Protecció contra les tècniques de falsificació de petició en llocs creuats (*cross-site request forgery*, CSRF), que són un tipus d'atac amb el que s'utilitza una persona usuària amb la sessió iniciada per fer accions no permeses en el lloc web. S'utilitza la protecció de *Laravel* per defecte col·locant en tots els formularis del lloc web el comandament `@csrf`, que serveix per enviar un *token* de verificació que permet prevenir un possible atac [29].
- Validació en els formularis, és a dir, afegir condicions que cal que cada camp d'un formulari compleixi abans de processar la informació que conté. Això es fa per una banda en el mateix formulari, per a que el propi navegador en faci una primera comprovació, i després a través dels controladors, afegint funcions de validació. S'utilitza per exemple per limitar el nombre de caràcters d'un camp, demanar que s'hagi omplert o comprovar que una adreça electrònica té el format que ha de tenir.
- Evitar la injecció de codi SQL, que afectaria a la base de dades, amb el constructor de peticions a la base de dades de *Laravel*, que utilitza *PDO parameter binding* (enllaç de paràmetres PDO) [15].
- Prevenció d'atacs XSS (*Cross-site scripting*, execució de comandaments en llocs creuats). Utilitzant *Blade*, escrivint `{{ }}` en els camps on s'introdueix i es mostra la informació en els formularis, es fa una crida a la funció *htmlspecialchars* de PHP [30]. Aquesta funció converteix els caràcters especials en entitats HTML [31].

10. Disseny de l'estil

Aquest treball s'ha orientat a la part estructural d'un lloc web. Tot i això, també s'han de tenir en compte alguns aspectes estètics, ja que influeixen en la funcionalitat del lloc web i per tant en que es compleixin els seus objectius.

10.1. Arxius css d'estil

L'estil, és a dir, els colors, els fonts de lletra, les mides i la col·locació dels diferents elements dins la pàgina, etc. es guarden en un arxiu separat, en llenguatge css. S'ha creat un sol arxiu propi per a l'estil, que sobreescriu el de *Bootstrap* per poder personalitzar-lo. Per al panell d'administració s'ha deixat l'estil que venia per defecte amb *Voyager*, per tant per a aquesta part del lloc web no s'aplica el css personalitzat que hi ha a la resta de pàgines. S'ha considerat que l'estil per defecte ja complia els requisits d'usabilitat i que no afectava a la imatge de marca perquè només és visible per a les persones amb alguns permisos d'administració.

10.2. Disseny visual

Alguns aspectes destacats en el disseny visual són els següents:

- Adaptable a diferents mides de pantalla i dispositius. Per a aconseguir això, s'ha utilitzat *Bootstrap 3*, tal com s'ha indicat en capítols anteriors.
- Ús d'icones en diferents punts del lloc web. Les icones faciliten la comprensió visual. S'han utilitzat icones provinents de la llibreria de *Bootstrap 3*, de *Voyager* i de llibreries externes, aquestes últimes per a les xarxes socials.
- Accessibilitat. Tot i que es podria millorar aquest aspecte, s'ha volgut seguir les recomanacions d'accessibilitat de *W3C* [32].
- Simplicitat a l'estructura i a la visualització, evitant elements innecessaris.
- Claredat a les explicacions legals, concretament les relacionades amb la normativa de protecció de dades.
- Taula de colors basada en el logotip.

11. Programari utilitzat

A més de les eines de programació explicades en el capítol 4, s'ha utilitzat programari complementari.

11.1. Editor de codi

Un bon editor de codi facilita la tasca d'escriure el codi amb diverses funcionalitats, com la distinció per colors dels elements del codi, treballar veient l'estructura de carpetes, el tancament automàtic d'etiquetes quan s'escriu la obertura, permetre l'ús de combinacions de tecles per les funcionalitats més comuns, facilita trobar arxius relacionats amb l'actual mentre es treballa, etc.

S'ha triat *Sublime Text 3* [33] com a editor. No és gratuït, però té un període de prova indefinit. Per a ús continuat cal comprar una llicència, que costa 80\$.

11.2. Administrador de bases de dades

Tot i que amb el panell d'administració de *Voyager* en principi es pot fer tota la gestió de la base de dades i les seves taules, durant el procés de creació del lloc web és útil tenir una eina addicional més específica. S'ha utilitzat *phpMyAdmin* [34].

És de codi obert, està dirigit a bases de dades MySQL, té un equip que fa les feines de manteniment i actualització i hi ha una àmplia xarxa de persones usuàries que poden donar suport en cas que hi hagi algun error. Està escrit en PHP i per tant cal tenir-lo instal·lat i en funcionament per utilitzar *phpMyAdmin*. Permet treballar amb dues modalitats: en consola i a través de la interfície.

Té una vista general de totes les bases de dades creades, que es poden obrir per veure'n més detalls segons les necessitats i a més té altres vistes, per l'estructura de cada taula, pel contingut de la taula sencera, pel contingut d'una entrada en particular, per la configuració, etc. Es poden buscar, editar o esborrar tant bases de dades, com taules com elements de les taules; o bé canviar-ne l'estructura de cada una.

A més també es poden escriure comandes SQL directament, tant a la consola com a la interfície. Això fa que sigui molt més adaptable que només un panell d'administració. Permet importar arxius en format CSV i SQL i exportar-los en més formats.

11.3. Fals servidor de correu electrònic

Per fer les proves d'enviar correus electrònics des del lloc web, s'ha utilitzat *Mailtrap* [35], que és un fals servidor. Utilitzar aquest servidor permet fer les proves treballant en un entorn local (no a internet) i a més evita un enviament massiu de correus d'assaig. Ajudar a verificar que els correus s'arriben a enviar i que ho fan amb el text i el format desitjat.

Mailtrap té diferents preus segons el servei que ofereix. S'ha triat la opció gratuïta perquè les proves que calia fer eren poques i només per a un lloc web.

11.4. Programari de suport a *Laravel*

Per fer servir *Laravel* convé també utilitzar altres programes.

Git [36] és un programari de control de versions. S'utilitza com a suport a la gestió de tot el que conté el projecte. És gratuït i de codi obert.

Composer [37] és una eina de gestió de dependències en PHP. Ajuda a controlar les llibreries instal·lades en el projecte i les relacions entre elles. Es distribueix lliurement sota una llicència MIT.

VirtualBox [38] permet generar un entorn virtual per a treballar amb el projecte. Està disponible gratuïtament.

Vagrant [39] s'utilitza per crear i configurar l'entorn de treball virtual. També és de codi obert.

Laravel Homestead [40] és un conjunt de programari pensat per utilitzar *Vagrant* i preparar l'entorn de treball per a *Laravel*, instal·lant PHP, *MySQL* i la resta de programari necessari.

12. Pressupost

La duració d'aquest treball ha estat d'unes vint setmanes. Els costos derivats de realitzar-lo es poden dividir en directes i indirectes. Els costos directes es poden comptar a partir de les hores necessàries de dedicació i el consum d'electricitat, mentre que els indirectes es corresponen amb l'amortització de l'ordinador utilitzat i els desplaçaments necessaris. No s'han pagat llicències ni s'ha comprat material directament per a aquest projecte.

De tots aquests costos només es tindrà en compte el corresponent al salari i la resta es consideraran negligibles, ja que hi ha dos o més ordres de magnitud de diferència entre els seus valors.

Per al càlcul d'hores es compta que s'han dedicat vint setmanes, amb una mitjana de vint hores setmanals per a la realització d'aquest treball. En aquest temps es compten les tasques de recerca d'informació, aprenentatge de les tecnologies a utilitzar, procés d'instal·lació de tot el programari necessari, desenvolupament del codi i redacció de la memòria. A més s'afegiran dues setmanes més per a la preparació de la presentació.

En total, 22 setmanes x 20 hores/setmana = 440 hores.

Per calcular el cost salarial, s'ha valorat un cost horari de 10 €/h per a una enginyera en fase de formació.

Concepte	Cost horari	Hores	Cost total
Salari	10 €/h	440 hores	4400 €
Total			4400 €

Taula 12.1: pressupost

El pressupost total és de 4400 €.

Conclusions

Durant la realització d'aquest treball s'ha creat un lloc web funcional que compleix els requisits marcats. Podrà ser utilitzat per l'equip del col·lectiu per donar-se a conèixer i promocionar els seus serveis i les seves activitats. A més, també l'ajudarà a gestionar la informació relacionada amb aquestes últimes. Les persones usuàries de l'espai podran proposar i inscriure's a les activitats, anunciar-se a través del directori i gestionar la seva informació.

S'ha desenvolupat una estructura per a les pàgines que formen el lloc web. A més, també s'han estructurat els arxius de les vistes i s'ha aprofitat l'estructura de carpetes generada per el programari instal·lat. S'ha decidit utilitzar *Laravel* com a entorn de treball i *Voyager* per a la creació d'un panell d'administració.

S'ha creat una base de dades amb diverses taules relacionades entre si. També s'han creat els arxius que permeten interaccionar amb la base de dades, introduir-hi nova informació, tractar-la i mostrar-ne els resultats. A més s'ha generat el sistema de rutes per enllaçar els arxius entre si.

Finalment, per a tot això, s'ha respectat la legalitat vigent, s'han tingut en compte aspectes visuals, s'ha utilitzat programari de suport i s'ha avaluat el pressupost necessari per a la realització d'aquest treball.

Agraïments

A les persones que formen l'equip de treball de La Trobada, gràcies pel suport ofert, pels suggeriments i per la paciència. Gràcies també al tutor d'aquest treball, l'Oriol Boix, per l'acompanyament, les propostes de millora i la bona disposició per ajudar.

Bibliografia

Referències bibliogràfiques

1. 2018 reform of EU data protection rules | European Commission. A: *European Commission* [en línia]. [Consulta: 10 juny 2018]. Disponible a: https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en.
2. Agencia Estatal del Boletín Oficial del Estado. Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. A: *BOE* [en línia]. 1999. DOI BOE-A-1999-23750. [Consulta: 10 juny 2018]. Disponible a: <https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750>.
3. Agencia Estatal del Boletín Oficial del Estado. Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal. A: *BOE* [en línia]. 2008. DOI BOE-A-2008-979. [Consulta: 10 juny 2018]. Disponible a: <https://www.boe.es/buscar/act.php?id=BOE-A-2008-979>.
4. Agencia Estatal del Boletín Oficial del Estado. Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico. A: *BOE* [en línia]. 2002. DOI BOE-A-2002-13758. [Consulta: 10 juny 2018]. Disponible a: <https://www.boe.es/buscar/act.php?id=BOE-A-2002-13758>.
5. Otwell, T. Laravel - The PHP Framework For Web Artisans. A: [en línia]. [Consulta: 10 juny 2018]. Disponible a: <https://laravel.com/>.
6. The Control Group. Voyager - The Missing Laravel Admin. A: [en línia]. 2018. [Consulta: 10 juny 2018]. Disponible a: <https://laravelvoyager.com/>.
7. Otto, M. i Fat. Bootstrap · The most popular HTML, CSS, and JS library in the world. A: [en línia]. [Consulta: 10 juny 2018]. Disponible a: <https://getbootstrap.com/>.
8. Otto, M. i Fat. Bootstrap 3.3. A: [en línia]. [Consulta: 10 juny 2018]. Disponible a: <https://getbootstrap.com/docs/3.3/>.
9. Formcarry. | no Javascript no PHP just point your form and chill. A: [en línia]. [Consulta: 10 juny 2018]. Disponible a: <https://www.formcarry.com/>.
10. Otwell, T. Authentication - Laravel. A: *Laravel* [en línia]. [Consulta: 10 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/authentication>.
11. Hernandez, U. MVC (Model, View, Controller) explicado. A: [en línia]. [Consulta: 13 abril 2018]. Disponible a: <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>.
12. Otwell, T. Middleware - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/middleware>.

13. Lloc web oficial de Google Analytics - Anàlisi web i informes – Google Analytics. A: *Google* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <http://www.google.es/intl/ca/analytics/>.
14. Oracle Corporation. MySQL :: MySQL Community Downloads. A: *Oracle* [en línia]. 2018. [Consulta: 11 juny 2018]. Disponible a: <https://dev.mysql.com/downloads/>.
15. Otwell, T. Database: Getting Started - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/database>.
16. Otwell, T. Eloquent: Getting Started - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/eloquent>.
17. Otwell, T. Eloquent: Serialization - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/eloquent-serialization#date-serialization>.
18. Nesbitt, B. Carbon - A simple PHP API extension for DateTime. A: [en línia]. [Consulta: 11 juny 2018]. Disponible a: <http://carbon.nesbot.com/>.
19. Otwell, T. Notifications - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/notifications>.
20. Otwell, T. Routing - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/routing>.
21. Otwell, T. Authorization - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/authorization#creating-policies>.
22. Otwell, T. Controllers - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/controllers>.
23. Otwell, T. Blade Templates: Control Structures - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/blade#control-structures>.
24. The Control Group. Roles and Permissions: Using Permissions in your Blade Template files - Voyager. A: *Voyager* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://voyager.readme.io/docs/permissions#section-using-permissions-in-your-blade-template-files>.
25. Google. Google Maps. A: [en línia]. 2018. [Consulta: 11 juny 2018]. Disponible a: <https://www.google.com/maps>.
26. Mozilla y colaboradores. Contraseñas Inseguras - Seguridad. A: *MDN* [en línia]. 2018. [Consulta: 11 juny 2018]. Disponible a: <https://developer.mozilla.org/es/docs/Seguridad/ContraseñasInseguras>.
27. Vyas, T. No More Passwords over HTTP, Please! A: *blog.mozilla.org* [en línia]. 2016. [Consulta: 6 juny 2018]. Disponible a: <https://blog.mozilla.org/tanvi/2016/01/28/no-more-passwords-over-http-please/>.
28. Otwell, T. Authentication: Login Throttling - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/authentication#login-throttling>.

29. Otwell, T. CSRF Protection - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/csrf>.
30. Otwell, T. Blade Templates: Displaying Data - Laravel. A: *Laravel* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/blade#displaying-data>.
31. The PHP Group. PHP: htmlspecialchars - Manual. A: *php.net* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <http://php.net/manual/en/function htmlspecialchars.php>.
32. Chisholm, W., Vanderheiden, G. i Jacobs, I. Web Content Accessibility Guidelines 1.0. A: *W3C* [en línia]. 1999. [Consulta: 11 juny 2018]. Disponible a: <https://www.w3.org/TR/WAI-WEBCONTENT/>.
33. Sublime Text - A sophisticated text editor for code, markup and prose. A: [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://www.sublimetext.com/>.
34. phpMyAdmin. A: [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://www.phpmyadmin.net/>.
35. Mailtrap.io — Fake smtp testing server. Dummy smtp email testing. A: [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://mailtrap.io/>.
36. Git - Downloads. A: [en línia]. [Consulta: 6 juny 2018]. Disponible a: <https://git-scm.com/downloads>.
37. Composer. A: [en línia]. [Consulta: 6 juny 2018]. Disponible a: <https://getcomposer.org/download/>.
38. Oracle VM VirtualBox. A: [en línia]. [Consulta: 6 juny 2018]. Disponible a: <https://www.virtualbox.org/>.
39. HashiCorp. Vagrant. A: [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://www.vagrantup.com/>.
40. Laravel Homestead - Laravel - The PHP Framework For Web Artisans. A: [en línia]. [Consulta: 6 juny 2018]. Disponible a: <https://laravel.com/docs/5.6/homestead>.

Bibliografia complementària

- Jacurtis., 2016. How to Build a Blog with Laravel - Part 1 (Exploring File Structure) - DevMarketer. A: *Youtube* [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://www.youtube.com/watch?v=R8B4og-BeCk&list=PLwAKR305CRO-Q90J---jXVzbOd4CDRbVx&index=1>.
- Miríadax., 2018. Miríadax - Curso MOOC de desarrollo de APPS HTML5. A: [en línia]. [Consulta: 15 març 2018]. Disponible a: <https://miriadax.net/web/html5mooc/inicio?timestamp=>.
- Píldoras Informáticas – Cursos gratuitos de informática. A: [en línia]. [Consulta: 11 juny 2018]. Disponible a: <http://www.pildorasinformaticas.es/>.

Privacy Policy Formcarry. A: [en línia]. [Consulta: 12 abril 2018]. Disponible a: <https://www.formcarry.com/privacy>.

Terms of Service Formcarry. A: [en línia]. [Consulta: 12 abril 2018]. Disponible a: <https://www.formcarry.com/terms>.

W3Schools Online Web Tutorials. A: [en línia]. [Consulta: 11 juny 2018]. Disponible a: <https://www.w3schools.com/>.

Annex

A.1. Rutes

```
<?php

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/
use App\Activitat;
use App\Director;

Route::get('/', 'IndexController')->name('index');
Route::get('/index', 'IndexController');

Auth::routes(); //rutes del sistema de registre i d'inici de sessió

Route::get('/home', 'HomeController@index')->name('home');

Route::group(['prefix' => 'admin'], function () {
    Voyager::routes();

    // Your overwrites here
    Route::post('logout', ['uses' => 'Auth\LoginController@logout', 'as'
=> 'voyager.logout']);
});

//rutes secció de presentació

Route::get('/projecte', function() {
    return view('presentacio/projecte');
});

Route::get('/equip', function() {
    return view('presentacio/equip');
});

Route::get('/onsom', function() {
    return view('presentacio/onsom');
});

Route::get('/contacte', function() {
    return view('presentacio/contacte');
})->name('contacte');
```

```

//rutes secció de serveis

Route::get('/espais', function() {
    $sales = App\Sales::all();
    return view('serveis/espais', compact('sales'));
});

Route::get('/cotreball', function() {
    return view('serveis/cotreball');
});

Route::get('/exposicions', function() {
    return view('serveis/exposicions');
});

//ruta condicions d'ús i política de privacitat
Route::get('/condicions-us', function() {
    return view('normativa');
})->name('condicions');

//rutes articles
Route::get('/post/{slug}', 'PostController@show');
Route::get('/articles', 'PostController@index');

//rutes activitats
Route::get('/activitat/{slug}', array('as' => 'showActivitat', 'uses' =>
'ActivitatController@show'));
Route::get('/activitats', array('as' => 'activitats', 'uses' =>
'ActivitatController@index'));
Route::get('/crear-activitat', array('as' => 'crear-activitat', 'uses' =>
'ActivitatController@create'))->middleware('auth');
Route::post('/store-activitat/{id_user}', array('as' => 'store-activitat',
'uses' => 'ActivitatController@store'))->middleware('auth');
Route::get('/editar-activitat/{activitat}', array('as' => 'editar-
activitat', 'uses' => 'ActivitatController@edit'))->
middleware('can:update,activitat');
Route::put('/update-activitat/{activitat}', array('as' => 'update-
activitat', 'uses' => 'ActivitatController@update'));
Route::delete('/destruir-activitat/{activitat}', array('as' => 'destruir-
activitat', 'uses' => 'ActivitatController@destroy'));
Route::patch('/attach-assistent/{id_act}/{id_user}', array('as'
=> 'attach-assistent', 'uses' => 'ActivitatController@atassistent'));
Route::patch('/detach-assistent/{id_act}/{id_user}', array('as'
=> 'detach-assistent', 'uses' => 'ActivitatController@detassistent'));

// rutes directori
Route::get('/directori', array('as' => 'directori', 'uses' =>
'DirectoriController@index'));
Route::get('/crear-directori', array('as' => 'crear-directori', 'uses' =>
'DirectoriController@create'))->middleware('can:create,App\Directori');
Route::post('/store-directori/{id_user}', array('as' => 'store-directori',
'uses' => 'DirectoriController@store'));
Route::get('/editar-directori/{directori}', array('as' => 'editar-
directori', 'uses' => 'DirectoriController@edit'))->
middleware('can:update,directori');

```

```

Route::put('/update-directori/{directori}', array('as' => 'update-
directori', 'uses' => 'DirectoriController@update'));
Route::delete('/destruir-directori/{directori}', array('as' => 'destruir-
directori', 'uses' => 'DirectoriController@destroy'));

// rutes perfil
Route::post('/perfil', array('as' => 'perfil', 'uses' =>
'PerfilController@perfil'))->middleware('auth');
Route::post('/perfil#activitats', array('as' => 'perfil-activitats',
'uses' => 'PerfilController@perfil'))->middleware('auth');
Route::post('/perfil#directori', array('as' => 'perfil-directori', 'uses'
=> 'PerfilController@perfil'))->middleware('auth');
Route::get('/editar-perfil', array('as' => 'editar-perfil', 'uses' =>
'PerfilController@edit'))->middleware('auth');
Route::get('/esborra-perfil', array('as' => 'esborra-perfil', 'uses' =>
'PerfilController@esborra'))->middleware('auth');
Route::put('/update-perfil', array('as' => 'update-perfil', 'uses' =>
'PerfilController@update'))->middleware('auth');
Route::delete('/destruir-perfil', array('as' => 'destruir-perfil', 'uses'
=> 'PerfilController@destroy'))->middleware('auth');

```

A.2. Models

A.2.1. Activitat

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Carbon\Carbon;
use TCG\Voyager\Models;
use Illuminate\Database\Eloquent\SoftDeletes;

class Activitat extends Model
{
    use SoftDeletes;

    protected $dates = [
        'data_inici', 'data_final', 'data_limit_inscr',
        'created_at', 'updated_at', 'deleted_at'
    ];

    protected $casts = [
        'hora_inici' => 'time',
        'hora_final' => 'time',
    ];

    public function save(array $options = [])
    {
        // If no author has been assigned, assign the current user's id as
        the author of the post
    }

```



```

        if (!$this->direccio) {
            $this->direccio = setting('site.direccio');
        }

        parent::save();
    }

    public function assistents()
    {
        return $this->belongsToMany('App\User', 'activ_assistents',
            'activitat_id', 'user_id');
    }

    public function responsables()
    {
        return $this->belongsToMany('App\User', 'activitat_responsables',
            'activitat_id', 'user_id');
    }
}

```

A.2.2. Directori

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;

class Directori extends Model
{
    use SoftDeletes;

    protected $table = 'directori';

    protected $dates = [
        'created_at', 'updated_at', 'deleted_at'
    ];

    public function user()
    {
        return $this->belongsTo('App\User');
    }
}

```

A.2.3. Publics

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

```

```
class Publics extends Model
{
    protected $table = 'public';
}
```

A.2.4. Sales

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Sales extends Model {
}
```

A.2.5. User

```
<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;
use App\Notifications\ResetPasswordCat;

class User extends \TCG\Voyager\Models\User
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    public function sendPasswordResetNotification($token)
    {
        $this->notify(new ResetPasswordCat($token));
    }
}
```

```

        public function assistents()
        {
            return $this->belongsToMany('App\Activitat', 'activ_assistents');
        }

        public function responsables()
        {
            return $this->belongsToMany('App\Activitat',
            'activitat_responsables');
        }

        public function directori()
        {
            return $this->hasOne('App\Directori');
        }
    }
}

```

A.3. Vistes

A.3.1. index

```

@extends('layouts.plantilla')

@section('content')

<section class="col-md-8 column contingut">

    @if ($destacat !== null)
    <article class="destacat" style="background-image: url({{
'storage/destacats/' . $destacat->slug . '.jpg' }})">
        @if ($destacat->estat)
            <h2><a href="{{ route('showActivitat', $destacat->slug) }}">{{
            $destacat->titol }}</a></h2><br>
            <h3>{{ $destacat->data_inici->format('d/m/Y') }}</h3>
            <p>{!! $destacat->resum !!}</p>
        @elseif ($destacat->status)
            <h2><a href="{{ url('/post/' . $destacat->slug) }}">{{
            $destacat->title }}</a></h2>
            <p>{{ $destacat->excerpt }}</p>
        @endif
    </article>
    <br>
    @endif

    <div class="row">
    @foreach($activitats as $activitat)
        <article class="col-md-6">
            @if ($activitat->imatge !== null)
            <br>
            @endif
            <h4>
                {{ $activitat->titol }}
            </h4>
        </article>
    @endforeach
    </div>

```

```

<p>{{ $activitat->data_inici->format('d/m/Y') }}</p>
<p>{!! $activitat->resum !!}</p>
<p><a href="{{ route('showActivitat', $activitat->slug) }}">{{
__('Més informació') }}</a></p><br>
</article><br>
@endforeach
@foreach ($posts as $post)
<article class="col-md-6">
    @if ($post->image !== null)
    <br>
    @endif
    <h4>
        {{ $post->title }}
    </h4>
    <p>{{ $post->excerpt }}</p>
    <p class="list-group-item-text"><a href="{{ url('/post/' .
    $post->slug) }}">{{ __('Seguir llegint') }}</a></p><br>
</article><br>
@endforeach
</div>
</section>

<section class="col-md-4 column lateral">
<aside>
<article>
    <h4>On trobar-nos</h4>
    <div class="embed-responsive embed-responsive-16by9">
        <iframe class="embed-responsive-item" width="100%"
        frameborder="0" scrolling="no" marginheight="0" marginwidth="0" src="..."
        style="border:0" allowfullscreen></iframe>{{-- mapa --}}
    </div>
    <p>{{ setting('site.direccio') }}</p>
    <p>Hi som de 10 a 14h de dilluns a divendres</p>
</article><br>
<article>
    <h4>Envia'ns les teves propostes d'activitat</h4>
    <p>T'agradaria organitzar el teu propi curs? Et dediques a fer
    tallers? Necessites un espai per organitzar la teva taula rodona?
    <a href="{{ route('crear-activitat') }}">{{ __("Proposa'ns una
    activitat") }}</a></p>
</article><br>
<article>
    <h4>Vine al nostre espai de cotreball</h4>
    <p>Busques un espai on anar a treballar? T'oferim un entorn de
    treball dedicat a l'economia social i solidària i als moviments
    socials. <a href="{{ url('/cotreball') }}">{{ __("Consulta els
    detalls aquí") }}</a></p>
</article><br>
<article>
    <h4>Visita el nostre directori</h4>
    <p>Persones, col·lectius i entitats que treballen per l'economia
    social i solidària. <a href="{{ route('directori') }}">{{ __("Fem
    xarxa!") }}</a></p>
</article><br>
<article>
    <h4>Exposa amb nosaltres</h4>

```

```

    <p>Tenim un espai per a exposicions. Vine a veure-les o porta'ns la
    teva exposició. <a href="{{ url('/exposicions') }}">{{ __("Consulta
    els detalls aquí") }}</a></p>
  </article><br>
</aside>
</section>
@endsection

```

A.3.2. activitats/activundia

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . $activitat->titol)

@section('content')

<article>
  <div class="row">
    <h1 class="col-md-6">{{ $activitat->titol }}</h1>
    @include('activitats.partials.botons')
  </div>

  @if ($activitat->imatge !== null)
  <br>
  @endif

  <p>Tipus d'activitat: {{ $activitat->tipus_act }}</p>
  <p>Lloc: {{ $activitat->direccio }} - {{ $sala }}</p>
  <p>Dia i hora: {{ $activitat->data_inici->format('d/m/Y') }} a les
  {{ date("H:i", strtotime($activitat->hora_inici)) }}</p>
  <p>Hora de finalització: {{ $activitat->hora_final }}</p>
  <br>
  <p>{!! $activitat->descr_llarga !!}</p>
  <br>
  @if ($activitat->aquilla_inv == 'Sí')
  <p>Es farà taquilla inversa, és a dir, un cop finalitzada
  l'activitat pots decidir quina quantitat vols pagar. El preu
  recomenat és {{ $activitat->preu }}.</p>
  @else
  <p>Preu: {{ $activitat->preu }}</p>
  @endif
  <br>

  @include('activitats.partials.assistents')

</article>
<br>
<br>

@endsection

```

A.3.3. activitats/curs

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . $activitat->titol)

@section('content')

<article>
    <div class="row">
        <h1 class="col-md-6">{{ $activitat->titol }}</h1>
        @include('activitats.partials.botons')
    </div>
    @if ($activitat->imatge !== null)
    <br>
    @endif
    <p>Tipus d'activitat: {{ $activitat->tipus_act }}</p>
    <p>Lloc: {{ $activitat->direccio }} - {{ $sala }}</p><p></p>
    <p>Data d'inici: {{ $activitat->data_inici->format('d/m/Y') }}</p>
    <p>Data de finalització: {{ $activitat->data_final->format('d/m/Y')
    }}</p>
    <p>Horari: {{ $activitat->horari }}</p>
    <br>
    <p>{!! $activitat->descr_llarga !!</p>
    <br>
    <p>Preu: {{ $activitat->preu }}</p>
    <br>
    @include('activitats.partials.assistents')
</article>
<br>
<br>

@endsection

```

A.3.4. activitats/jornades

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . $activitat->titol)

@section('content')

<article>
    <div class="row" style="width: 100%">
        <h1 class="col-md-6">{{ $activitat->titol }}</h1>
        @include('activitats.partials.botons')
    </div>
    @if ($activitat->imatge !== null)
    <br>
    @endif
    <p>Direcció: {{ $activitat->direccio }}</p>

```

```

    <p>Data d'inici: {{ $activitat->data_inici->format('d/m/Y') }} a
    partir de les {{ date("H:i", strtotime($activitat->hora_inici))
    }}</p>
    <p>Data de finalització: {{ $activitat->data_final->format('d/m/Y')
    }} fins a les {{ date("H:i", strtotime($activitat->hora_final))
    }}</p>
    <br>
    <p>{!! $activitat->descr_llarga !!}</p>
    <br>
    <p>Preu: {{ $activitat->preu }}</p>
    <br>
    @include('activitats.partials.assistents')
</article>
<br>
<br>
@endsection

```

A.3.5. activitats/allactivitats

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Activitats
programades")

@section('content')

<article>
    <h1>Pròximes activitats</h1>
    <br>

    <ul class="list-group">
    @foreach($activitats as $activitat)
        <li class="list-group-item col-md-4">
            <h4 class="list-group-item-heading">{{ $activitat->titol }}</h4>

            @if ($activitat->imatge != null)
                <br>
            @endif

            <p>Tipus d'activitat: {{ $activitat->tipus_act }}</p>
            <p>Inici: {{ $activitat->data_inici->format('d/m/Y') }} a les {{
            date("H:i", strtotime($activitat->hora_inici)) }}</p>
            <p class="list-group-item-text">{!! $activitat->resum !!}</p>
            <p class="list-group-item-text"><a href="{{ route('showActivitat',
            $activitat->slug) }}">{{ __( 'Més informació' ) }}</a></p>

        </li>
    @endforeach
    </ul>

    <br>

</article>

```

```
<br><br>

@endsection
```

A.3.6. activitats/senseactivitats

```
@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Activitats
programades")

@section('content')

<div class="row">
  <div class="col-md-8 col-md-offset-2">
    <h1>Actualment no hi ha activitats programades</h1>
  </div>
</div>

<br><br>

@endsection
```

A.3.7. activitats/crear

```
@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Proposar una nova
activitat")

@section('content')

<h1>Proposa una nova activitat</h1>
<br>
<p>Si hi ha més persones responsables, pots indicar-ho a la descripció.
Si hi ha les adreces electròniques amb les que s'han registrat, les
incorporarem a la supervisió perquè puguin revisar la informació de la
pàgina.</p>
<p>Totes les dades que consideris necessàries i no estiguin al formulari
afegeix-les a la descripció. Per exemple, si hi ha dates a les que es
pugui realitzar l'activitat o només pot ser a algunes hores, pots
indicar-ho allà. Pots deixar buits els camps que vulguis excepte el
títol. Si hi ha alguna incidència o necessites ajuda, pots posar-te <a
href="{{ route('contacte') }}">{{ __('en contacte amb nosaltres')
}}</a>.</p>
<br>

{{ Form::open(array('route' =>['store-activitat', Auth::user()->id],
'files' => true)) }}

    @csrf

    {{ Form::label('titol', "Títol*") }}
```



```

    {{ Form::text('titol', null, array('class' => 'form-control',
    'required')) }}
    @if ($errors->has('titol'))
    @foreach ($errors->get('titol') as $message)
        <span class="invalid-feedback">
            <strong>{{ $message }}</strong>
        </span>
    @endforeach
    @endif
    <br>

    {{ Form::label('imatge', "Imatge") }}
    {{ Form::file('imatge') }}
    @if ($errors->has('imatge'))
    @foreach ($errors->get('imatge') as $message)
        <span class="invalid-feedback">
            <strong>{{ $message }}</strong>
        </span>
    @endforeach
    @endif
    <br>

    {{ Form::label('tipus_act', "Tipus d'activitat") }}
    {{ Form::select('tipus_act',[
        'Activitat' => 'Activitat',
        'Taller' => 'Taller',
        'Xerrada' => 'Xerrada',
        'Curs' => 'Curs',
        'Taula rodona': 'Taula rodona',
        'Jornades' => 'Jornades',
        'Altres' => 'Altres'
    ]) }}
    <br><br>

    {{ Form::label('data_inici', "Proposta de data o de data d'inici")
    }}
    {{ Form::date('data_inici', null, array('class' => 'form-control'))
    }}
    @if ($errors->has('data_inici'))
    @foreach ($errors->get('data_inici') as $message)
        <span class="invalid-feedback">
            <strong>{{ $message }}</strong>
        </span>
    @endforeach
    @endif
    <br>

    {{ Form::label('data_final', "Data de finalització") }} <small>(Si
    és una activitat de més d'un dia)</small>
    {{ Form::date('data_final', null, array('class' => 'form-control'))
    }}
    @if ($errors->has('data_final'))
    @foreach ($errors->get('data_final') as $message)
        <span class="invalid-feedback">
            <strong>{{ $message }}</strong>
        </span>
    @endforeach
    @endif

```

```

<br>

{{ Form::label('horari', "Proposta d'horari") }} <small>(Només per
als cursos)</small>
{{ Form::text('horari', null, array('class' => 'form-control')) }}
@if ($errors->has('horari'))
@foreach ($errors->get('horari') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('places_max', "Número màxim de places") }}
{{ Form::number('places_max', null, array('class' => 'form-
control')) }}
@if ($errors->has('places_max'))
@foreach ($errors->get('places_max') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('taquilla_inv', "Vols que es faci taquilla
inversa?") }} <small>(És a dir, hi ha un preu recomenat, però cada
persona decideix quina quantitat posar)</small>
{{ Form::select('taquilla_inv', ['Sí' => 'Sí', 'No' => 'No']) }}
<br><br>

{{ Form::label('resum', "Resum") }} <small>Aquesta és la part que
es veurà en el llistat d'activitats</small>
{{ Form::textarea('resum', null, array('class' => 'form-control',
'maxlength' => '50')) }}
@if ($errors->has('resum'))
@foreach ($errors->get('resum') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('descr_llarga', 'Descripció')}} <small>Escriu aquí
la descripció extensa que es veurà a la pàgina de l'activitat i
tota la informació que consideris necessària i no estigui a la
resta d'apartats</small>
{{ Form::textarea('descr_llarga', null, array('class' => 'form-
control', 'maxlength' => '1000')) }}
@if ($errors->has('descr_llarga'))
@foreach ($errors->get('descr_llarga') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif

```

```

<br>

    {{ Form::checkbox('condicions', 'value') }}
    {{ Form::label('condicions', "Clicant en aquesta casella acceptes
les condicions d'ús.*") }} <small>Tota la informació enviada s'utilitzarà
únicament per a la gestió i la promoció de l'activitat. L'equip de {{
setting('site.title') }} haurà d'aprovar qualsevol proposta enviada i pot
decidir no acceptar-la en absolut, editar les dades per a la seva
publicació o posar-se en contacte amb la persona o persones responsables
per plantejar una nova proposta. Aquí pots consultar les <a href="{{
route('condicions') }}">{{ __("condicions d'ús") }}</a> més extensament.
</small>
    @if ($errors->has('condicions'))
    @foreach ($errors->get('condicions') as $message)
        <span class="invalid-feedback">
            <strong>{{ $message }}</strong>
        </span>
    @endforeach
    @endif
<br>

    {{ Form::submit("Enviar", ['class' => 'btn btn-success center-block
btn-lg']) }}

{{ Form::close() }}
<p>Els camps amb * són obligatoris</p>

<br><br>

@endsection

```

A.3.8. activitats/editar

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Editar: " .
$activitat->titol)

@section('content')

<h1>Edita una activitat</h1>
<br>
<p>Pots deixar buits els camps que vulguis excepte el títol. Si hi ha
alguna incidència o necessites ajuda, pots posar-te <a href="{{
route('contacte') }}">{{ __('en contacte amb nosaltres') }}</a>.</p>
<br>
{!! Form::open(array('route' => ['update-activitat', $activitat->id])) !!}

    @method('PUT')
    @csrf

    {{ Form::label('titol', "Títol*") }}
    {{ Form::text('titol', $activitat->titol, array('class' => 'form-
control', 'required')) }}
    @if ($errors->has('titol'))

```

```

@foreach ($errors->get('titol') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('imatge', "Imatge") }}
{{ Form::file('imatge') }}
@if ($errors->has('imatge'))
@foreach ($errors->get('imatge') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('tipus_act', "Tipus d'activitat") }}
{{ Form::select('tipus_act', [
    'Activitat' => 'Activitat',
    'Taller' => 'Taller',
    'Xerrada' => 'Xerrada',
    'Curs' => 'Curs',
    'Taula rodona' => 'Taula rodona',
    'Jornades' => 'Jornades',
    'Altres' => 'Altres'
], $activitat->tipus_act) }}
<br><br>

@if ($activitat->estat !== 'Publicada')
{{ Form::label('data_inici', "Proposta de data o de data d'inici") }}
{{ Form::date('data_inici', $activitat->data_inici, array('class'
=> 'form-control')) }}
@if ($errors->has('data_inici'))
@foreach ($errors->get('data_inici') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('data_final', "Data de finalització") }} <small>(Si
és una activitat de més d'un dia)</small>
{{ Form::date('data_final', $activitat->data_final, array('class'
=> 'form-control')) }}
@if ($errors->has('data_final'))
@foreach ($errors->get('data_final') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

```

```

{{ Form::label('horari', "Proposta d'horari") }} <small>(Només per
als cursos)</small>
{{ Form::text('horari', $activitat->horari, array('class' => 'form-
control')) }}
@if ($errors->has('horari'))
@foreach ($errors->get('horari') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>
@endif

{{ Form::label('places_max', "Número màxim de places") }}
{{ Form::number('places_max', $activitat->places_max, array('class'
=> 'form-control')) }}
@if ($errors->has('places_max'))
@foreach ($errors->get('places_max') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('taquilla_inv', "Vols que es faci taquilla
inversa?") }} <small>(És a dir, hi ha un preu recomenat, però cada
persona decideix quina quantitat posar)</small>
{{ Form::select('taquilla_inv', ['Sí' => 'Sí', 'No' => 'No'],
$activitat->taquilla_inv) }}
<br><br>

{{ Form::label('resum', "Resum") }} <small>Aquesta és la part que
es veurà en el llistat d'activitats</small>
{{ Form::textarea('resum', $activitat->resum, array('class' =>
'form-control', 'maxlength' => '400')) }}
@if ($errors->has('resum'))
@foreach ($errors->get('resum') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('descr_llarga', 'Descripció')}} <small>Escriu aquí
la descripció extensa que es veurà a la pàgina de l'activitat i
tota la informació que consideris necessària i no estigui a la
resta d'apartats</small>
{{ Form::textarea('descr_llarga', $activitat->descr_llarga,
array('class' => 'form-control', 'maxlength' => '1000')) }}
@if ($errors->has('descr_llarga'))
@foreach ($errors->get('descr_llarga') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach

```

```

@endif
<br>

{{ Form::checkbox('condicions', null) }}
{{ Form::label('condicions', "Clicant en aquesta casella acceptes
les condicions d'ús.*") }} <small>Tota la informació enviada
s'utilitzarà únicament per a la gestió i la promoció de
l'activitat. L'equip de {{ setting('site.title') }} pot decidir no
acceptar la proposta enviada, editar les dades per a la seva
publicació o posar-se en contacte amb la persona o persones
resposables per plantejar una nova proposta. Aquí pots consultar
les <a href="{{ route('condicions') }}">{{ __("condicions d'ús")
}}</a> més extensament. </small>
@if ($errors->has('condicions'))
@foreach ($errors->get('condicions') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::submit("Enviar", ['class' => 'btn btn-success center-block
btn-lg']) }}

{{ Form::close() }}
<p>Els camps amb * són obligatoris</p>
@if($missatge !== null)
    <div class="row">
        <div class="col-md-8 col-md-offset-2 alert alert-success">
            {{ $missatge }}
        </div>
    </div>
@endif

<br><br>

@endsection

```

A.3.9. activitats/partials/botons

```

@can('update', $activitat)
    <div class="col-md-3">
        {{ Form::open(array('route' => ['editar-activitat',
$activitat])) }}

        @method('GET')
        @csrf

        {{ Form::submit("Edita", ['class' => 'btn btn-info']) }}

    {{ Form::close() }}
    <br>
    </div>
    <div class="col-md-3">

```

```

        {{ Form::open(array('route' =>['destruir-activitat',
        $activitat])) }}

        @method('DELETE')
        @csrf

        {{ Form::submit("Esborra", ['class' => 'btn btn-danger']) }}

        {{ Form::close() }}
    </div>
@endcan

```

A.3.10. activitats/partials/assistents

```

<p>
    @if ($assistents->count() == 1)
        S'ha apuntat una persona
    @else
        S'han apuntat {{ $assistents->count() }} persones
    @endif
</p>

@if (Auth::guest())

<p>Registra't o inicia sessió per apuntar-te</p>

@else

@foreach ($responsables as $responsable)
    @if ($responsable->id == Auth::id())
        <p>Assistents:
            @foreach($assistents as $assistent)
                {{ $assistent->name }}
            @endforeach
        </p>
    @endif
@endforeach

@php

$comprova = true;

@endphp

@foreach ($assistents as $assistent)

    @if ($assistent->id == Auth::id())

        {{ Form::open(array('route' => ['detach-assistent',
        $activitat->id, Auth::id()])) }}

        @method('PATCH')
        @csrf
    
```

```

        {{ Form::submit("Ja no vull assistir", ['class' => 'btn
        btn-danger center-block btn-lg']) }}

    {{ Form::close() }}

    @php $comprova = false; @endphp

    @endif

@endforeach

@if ($comprova == true)

    {{ Form::open(array('route' =>['attach-assistent', $activitat->id,
    Auth::id()])) }}

        @method('PATCH')
        @csrf

        {{ Form::submit("Apunta'm!", ['class' => 'btn btn-success
        center-block btn-lg']) }}

    {{ Form::close() }}

@endif

@endif

```

A.3.11. articles/allposts

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Notícies i
articles")

@section('content')

<article>
    <h1>Últims articles publicats</h1>
    <br>

    <ul class="list-group">
    @foreach($posts as $post)
        <li class="list-group-item col-md-4">
            <h4 class="list-group-item-heading">{{ $post->title }}</h4><br>
            @if ($post->image !== null)
                <br>
            @endif
            <p class="list-group-item-text">{{ $post->excerpt }}</p>
            <p class="list-group-item-text"><a href="{{ url('/post/' . $post-
            >slug) }}">{{ __('Seguir llegint') }}</a></p>
        </li>
    @endforeach
    </ul>

```



```

</article>

<br><br>

@endsection

```

A.3.12. articles/post

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . $post->title)

@section('content')

<article>
    <h1>{{ $post->title }}</h1><br>
    @if ($post->image !== null)
        <br>
    @endif
    <p>{!! $post->body !!}</p>
    <br>
    <p>Escrit per {{ $author }}</p>
</article>
<br>
<br>

@endsection

```

A.3.13. directori/index

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Directori: Fem
xarxa!")

@section('content')

<article>
    <h1>Directori</h1>
    <p>En aquest directori trobaràs persones, grups i entitats que
treballen dins dels principis de l'economia social i solidària.</p>
    @guest
        <p>Vols afegir la teva pròpia entrada? <a href="{{ route('register')
}}">{{ __("Registra't") }}</a> o <a href="{{ route('login') }}">{{
__("inicia sessió!") }}</a></p>
    @endguest
    @auth
        <p>Pots editar o eliminar la teva entrada </p>
        <div class="row">
            <div class="col-md-2 col-xs-2">

```

```

    {{ Form::open(array('route' => ['editar-directori',
Auth::user()->directori])) }}

    @method('GET')
    @csrf

    {{ Form::submit("Editar", ['class' => 'btn btn-info']) }}

    {{ Form::close() }}
</div>
<div class="col-md-2 col-xs-2 col-xs-offset-2">
    {{ Form::open(array('route' => ['destruir-directori',
Auth::user()->directori])) }}

    @method('DELETE')
    @csrf

    {{ Form::submit("Eliminar", ['class' => 'btn btn-danger']) }}

    {{ Form::close() }}
</div>
</div>

@else
<p>Vols afegir la teva pròpia entrada?</p>
{{ Form::open(array('route' => ['crear-directori'])) }}

    @method('GET')
    @csrf

    {{ Form::submit("Nova entrada", ['class' => 'btn btn-success']) }}

    {{ Form::close() }}
@endif
@endauth
<br>
<h3>Persones</h3>
<br>
<div class="row" style="width: 100%">
    <ul class="list-group">
        @foreach($entrades as $entrada)
            @if($entrada->tipus == 'PERSONA')
                <li class="list-group-item col-md-4">
                    <h4 class="list-group-item-heading"> {{ $entrada->nom
                    }}</h4><br>
                    <br>
                    <p class="list-group-item-text">Web: <a href="{{
                    $entrada->web }}">{{ $entrada->web }}</a></p>
                    <p class="list-group-item-text">Adreça electrònica: {{
                    $entrada->email }}</p>
                    <p class="list-group-item-text">Adreça postal: {{
                    $entrada->adreca }}</p>
                    <br>
                    <p class="list-group-item-text">{{ $entrada->descripcio
                    }}</p>
                </li>
            @endif
        @endforeach
    </ul>
</div>

```

```

    @endforeach
</ul>
</div>
<br>
<h3>Grups, col·lectius i entitats</h3>
<br>
<div class="row" style="width: 100%">
  <ul class="list-group">
    @foreach($entrades as $entrada)
      @if($entrada->tipus == 'ENTITAT')
        <li class="list-group-item col-md-4">
          <h4 class="list-group-item-heading">{{ $entrada->nom
          }}</h4><br>
          <br>
          <p class="list-group-item-text">Web: <a href="{{
          $entrada->web }}">{{ $entrada->web }}</a></p>
          <p class="list-group-item-text">Adreça electrònica: {{
          $entrada->email }}</p>
          <p class="list-group-item-text">Adreça postal: {{
          $entrada->adreca }}</p>
          <br>
          <p class="list-group-item-text">{{ $entrada->descripcio
          }}</p>
        </li>
      @endif
    @endforeach
  </ul>
</div><br>

</article>

<br><br>

@endsection

```

A.3.14. directori/senseentrades

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Directori")

@section('content')

<div class="row">
  <div class="col-md-8 col-md-offset-2">
    <h1>Actualment no hi ha cap entrada en el directori</h1>
  </div>
</div>

<br><br>

@endsection

```

A.3.15. directori/nova-entrada

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Nova entrada per
al directori")

@section('content')

<h1>Nova entrada per al directori</h1>
<p>Amb aquest formulari pots crear la teva entrada. Omple els camps amb
els valors que vulguis que es mostrin. <strong>Important! Només es permet
una entrada per persona.</strong></p>
<br>

{{ Form::open(array('route' =>['store-directori', Auth::user()->id],
'files' => true)) }}

    @csrf

    {{ Form::label('nom', "Nom*:") }}
    {{ Form::text('nom', null, array('class' => 'form-control',
'required')) }}
    @if ($errors->has('nom'))
    @foreach ($errors->get('nom') as $message)
        <span class="invalid-feedback">
            <strong>{{ $message }}</strong>
        </span>
    @endforeach
    @endif
    <br>

    {{ Form::label('web', "Web (url):") }}
    {{ Form::text('web', null, array('class' => 'form-control')) }}
    @if ($errors->has('web'))
    @foreach ($errors->get('web') as $message)
        <span class="invalid-feedback">
            <strong>{{ $message }}</strong>
        </span>
    @endforeach
    @endif
    <br>

    {{ Form::label('tipus', "Tipus d'entrada") }}
    {{ Form::select('tipus_act',[
        'Persona' => 'PERSONA',
        'Entitat' => 'ENTITAT'
    ]) }}
    <br>

    {{ Form::label('email', "Adreça electrònica:") }}
    {{ Form::text('email', null, array('class' => 'form-control')) }}
    @if ($errors->has('email'))
    @foreach ($errors->get('email') as $message)
        <span class="invalid-feedback">
            <strong>{{ $message }}</strong>
        </span>
    @endforeach

```

```

@endif
<br>

{{ Form::label('adreca', "Adreça postal:") }}
{{ Form::text('adreca', null, array('class' => 'form-control')) }}
@if ($errors->has('adreca'))
@foreach ($errors->get('adreca') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('descripcio', 'Descripció:') }}
{{ Form::textarea('descripcio', null, array('class' => 'form-control')) }}
@if ($errors->has('descripcio'))
@foreach ($errors->get('descripcio') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('imatge', "Imatge:") }}
{{ Form::file('imatge') }}
@if ($errors->has('imatge'))
@foreach ($errors->get('imatge') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::checkbox('condicions', 'value') }}
{{ Form::label('condicions', "Clicant en aquesta casella acceptes les condicions d'ús.*") }} <small>Tota la informació enviada s'utilitzarà únicament per a la seva publicació a l'apartat del directori del lloc web. L'equip de {{ setting('site.title') }} haurà d'aprovar qualsevol proposta enviada i pot decidir no acceptar-la en absolut, editar les dades per a la seva publicació o posar-se en contacte amb la persona o persones responsables per plantejar una nova proposta. Aquí pots consultar les <a href="{{ route('condicions') }}">{{ __("condicions d'ús") }}</a> més extensament. </small>
@if ($errors->has('condicions'))
@foreach ($errors->get('condicions') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

```

```

        {{ Form::submit("Enviar", ['class' => 'btn btn-success center-block
        btn-lg']) }}

    {{ Form::close() }}
    <p>Els camps amb * són obligatoris</p>

    <br><br>

    @endsection

```

A.3.16. directori/editar

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Editar l'entrada
del directori")

@section('content')

    <h1>Editar una entrada</h1>
    <p>Amb aquest formulari pots editar la teva entrada. Omple els camps amb
    els valors que vulguis que es mostrin.</p>
    <br>

    {{ Form::open(array('route' => ['update-directori', $directori], 'files'
    => true)) }}

        @method('PUT')
        @csrf

        {{ Form::label('nom', "Nom*:") }}
        {{ Form::text('nom', $directori->nom, array('class' => 'form-
        control', 'required')) }}
        @if ($errors->has('nom'))
        @foreach ($errors->get('nom') as $message)
            <span class="invalid-feedback">
                <strong>{{ $message }}</strong>
            </span>
        @endforeach
        @endif
        <br>

        {{ Form::label('web', "Web (url):") }}
        {{ Form::text('web', $directori->web, array('class' => 'form-
        control')) }}
        @if ($errors->has('web'))
        @foreach ($errors->get('web') as $message)
            <span class="invalid-feedback">
                <strong>{{ $message }}</strong>
            </span>
        @endforeach
        @endif
        <br>

        {{ Form::label('email', "Adreça electrònica:") }}

```

```

{{ Form::text('email', $directori->email, array('class' => 'form-
control')) }}
@if ($errors->has('email'))
@foreach ($errors->get('email') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('adreca', "Adreça postal:") }}
{{ Form::text('adreca', $directori->adreca, array('class' => 'form-
control')) }}
@if ($errors->has('adreca'))
@foreach ($errors->get('adreca') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('descripcio', 'Descripció:') }}
{{ Form::textarea('descripcio', $directori->descripcio,
array('class' => 'form-control')) }}
@if ($errors->has('descripcio'))
@foreach ($errors->get('descripcio') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::label('imatge', "Imatge:") }}
{{ Form::file('imatge') }}
@if ($errors->has('imatge'))
@foreach ($errors->get('imatge') as $message)
    <span class="invalid-feedback">
        <strong>{{ $message }}</strong>
    </span>
@endforeach
@endif
<br>

{{ Form::checkbox('condicions', 'value') }}
{{ Form::label('condicions', "Clicant en aquesta casella acceptes
les condicions d'ús.*") }} <small>Tota la informació enviada
s'utilitzarà únicament per a la seva publicació a l'apartat del
directori del lloc web. L'equip de {{ setting('site.title') }}
haurà d'aprovar qualsevol proposta enviada i pot decidir no
acceptar-la en absolut, editar les dades per a la seva publicació o
posar-se en contacte amb la persona o persones responsables per
plantejar una nova proposta. Aquí pots consultar les <a href="{{
route('condicions') }}">{{ __("condicions d'ús") }}</a> més
extensament. </small>

```

```

@if ($errors->has('condicions'))
    @foreach ($errors->get('condicions') as $message)
        <span class="invalid-feedback">
            <strong>{{ $message }}</strong>
        </span>
    @endforeach
@endif
<br>

<br>

    {{ Form::submit("Enviar", ['class' => 'btn btn-success center-block btn-lg']) }}

{{ Form::close() }}
<p>Els camps amb * són obligatoris</p>

@if($missatge !== null)
    <div class="row">
        <div class="col-md-8 col-md-offset-2 alert alert-success">
            {{ $missatge }}
        </div>
    </div>
@endif

<br><br>

@endsection

```

A.3.17. perfil/perfil

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Perfil personal: " . $user->name)

@section('content')

<div class="row">
    <h1>Perfil</h1>
    <h3>Dades personals</h3>
    <div class="col-sm-6">
        <p>Aquestes són les teves dades</p>
        <p>Nom: {{ $user->name }}</p>
        <p>Adreça electrònica: {{ $user->email }}</p>
        <p>Rol: {{ $user->role->name }}</p>
        <br>
    </div>
    <div class="col-sm-3">
        <p>Avatar: </p>
        
    </div>
    <div class="col-sm-3">
        {{ Form::open(array('route' =>['editar-perfil'])) }}
    </div>

```



```

        @method('GET')
        @csrf

        {{ Form::submit("Edita", ['class' => 'btn btn-info']) }}

    {{ Form::close() }}
<br>
    {{ Form::open(array('route' =>['esborra-perfil'])) }}

    @method('GET')
    @csrf

    {{ Form::submit("Esborra", ['class' => 'btn btn-danger']) }}

    {{ Form::close() }}
</div>
</div>

<div class="row" id="activitats">
    <h3>Activitats</h3>

    @if((array_first($activresp) == null) && (array_first($activassis)
    == null))
        <p>No tens activitats en seguiment actualment.</p>
    @endif

    @if(array_first($activresp) != null)
        <p>Aquestes són les activitats de les que ets responsable</p>
        <ul class="list-group">
            @foreach($activresp as $activitat)
                <li class="list-group-item row">
                    <div class="col-sm-9">
                        <h4 class="list-group-item-heading">{{ $activitat->titol }}</h4>

                        <p class="list-group-item-text">Tipus d'activitat: {{
                        $activitat->tipus_act }}</p>
                        <p class="list-group-item-text">Inici: {{
                        $activitat->data_inici->format('d/m/Y') }} a les {{ date("H:i",
                        strtotime($activitat->hora_inici)) }}</p>
                        <p class="list-group-item-text">{!! $activitat->resum !!</p>
                        <p class="list-group-item-text"><a href="{{ url('/activitat/' .
                        $activitat->slug) }}">{{ __("Veure l'activitat") }}</a></p>
                    </div>
                    <div class="col-sm-3">
                        {{ Form::open(array('route' =>['editar-activitat', $activitat]))
                        }}

                        @method('GET')
                        @csrf

                        {{ Form::submit("Edita", ['class' => 'btn btn-info']) }}

                        {{ Form::close() }}
                        <br>
                        {{ Form::open(array('route' =>['destruir-activitat',
                        $activitat])) }}

                        @method('DELETE')

```

```

        @csrf

        {{ Form::submit("Esborra", ['class' => 'btn btn-danger']) }}

        {{ Form::close() }}
    </div>
</li>
@endforeach
</ul>
<br>
@endif
@if(array_first($activassis) != null)
    <p>Aquestes són les activitats a les que vols assistir</p>
    <ul class="list-group">
        @foreach($activassis as $activitat)
            <li class="list-group-item">
                <h4 class="list-group-item-heading">{{ $activitat->titol }}</h4>
                <p class="list-group-item-text">Tipus d'activitat: {{
                    $activitat->tipus_act }}</p>
                <p class="list-group-item-text">Inici: {{
                    $activitat->data_inici->format('d/m/Y') }} a les {{ date("H:i",
                    strtotime($activitat->hora_inici)) }}</p>
                <p class="list-group-item-text">{!! $activitat->resum !!</p>
                <p class="list-group-item-text"><a href="{{ url('/activitat/' .
                    $activitat->slug) }}">{{ __("Veure l'activitat") }}</a></p>
            </li>
        @endforeach
    </ul>
    <br>
</div>
@endif
@if($directori != null)
<div class="row" id="directori">
    <h3>Directori</h3>
    <p>Aquesta és la teva entrada en el directori</p>
    <ul class="list-group">
        <li class="list-group-item">
            <div class="col-sm-9">
                <h4 class="list-group-item-heading">{{ $directori->nom }}</h4>

                <p class="list-group-item-text">Tipus: {{ $directori->tipus
                }}</p>
                <p class="list-group-item-text">Web: {{ $directori->web }}</p>
                <p class="list-group-item-text">Adreça electrònica: {{
                    $directori->email }}</p>
                <p class="list-group-item-text">Adreça postal: {{
                    $directori->adrecas }}</p>
                <p class="list-group-item-text">Estat: {{ $directori->estat
                }}</p>
                <p class="list-group-item-text">Descripció: {{
                    $directori->descripcio }}</p>
            </div>
            <div class="col-sm-3">
                {{ Form::open(array('route' => ['editar-directori', $directori]))
                }}

                @method('GET')
                @csrf
            </div>
        </li>
    </ul>
</div>

```

```

        {{ Form::submit("Editar", ['class' => 'btn btn-info']) }}

        {{ Form::close() }}
    <br>
    {{ Form::open(array('route' => ['destruir-directori',
Auth::user()->directori])) }}

    @method('DELETE')
    @csrf

    {{ Form::submit("Eliminar", ['class' => 'btn btn-danger']) }}

    {{ Form::close() }}
</div>
</li>
</ul>
<br>
</div>
@endif
<br>
<br>
@endsection

```

A.3.18. perfil/esborrar

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Esborrar el
perfil de " . Auth::user()->name)

@section('content')

<br>
<div class="row">
    <div class="alert alert-danger col-md-8 col-md-offset-2"
        role="alert">
        <strong>Atenció!</strong> Segur que vols esborrar el teu
        compte? Es perdran les teves dades i no podràs accedir als
        serveis de {{ setting('site.title') }}. No és possible desfer
        aquest pas!
    </div>
</div>

<br>
<div class="row">
    <div class="col-md-1 col-md-offset-1 col-sm-4">
        {{ Form::open(array('route' => ['perfil'])) }}

        @csrf

        {{ Form::submit("Tornar a la pàgina de perfil",
            ['class' => 'btn btn-info']) }}
    </div>
</div>

```

```

        {{ Form::close() }}
    </div>
    <div class="col-md-1 col-md-offset-7 col-sm-4 col-sm-offset-2">
        {{ Form::open(array('route' =>['destruir-perfil'])) }}

        @method('DELETE')
        @csrf

        <div>
            <label for="password" class="col-form-label text-md-right">{{
                __('Contrasenya') }}</label>
            <input id="password" type="password" class="form-control"{{
                $errors->has('password') ? ' is-invalid' : '' }}"
                name="password" required>
            @if ($errors->has('password'))
                <span class="invalid-feedback">
                    <strong>{{ $errors->first('password') }}</strong>
                </span>
            @endif

        </div>

        {{ Form::submit("Esborra el meu compte", ['class' =>
            'btn btn-danger']) }}

        {{ Form::close() }}
    </div>
</div>
<br><br>

@endsection

```

A.3.19. perfil/editar

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Editar les dades
personals")

@section('content')
    <div class="container">
        <div class="row">
            <div class="col-md-8 col-md-offset-2">
                <div class="panel panel-default">
                    <div class="panel-heading">{{ __('Canviar les dades personals')
                    }}</div>

                    <div class="panel-body">
                        <form method="post" action="{{ route('update-perfil') }}"
                        enctype="multipart/form-data">
                            @method('PUT')
                            @csrf

                            <div class="form-group row">

```

```

<label for="name" class="col-md-4 col-form-label text-md-
right">{{ __('Nom') }}</label>

<div class="col-md-8">
  <input id="name" type="text" class="form-control{{
    $errors->has('name') ? ' is-invalid' : '' }}"
    name="name" value="{{ Auth::user()->name }}"
    placeholder="nom o pseudònim" required autofocus>

  @if ($errors->has('name'))
    <span class="invalid-feedback">
      <strong>{{ $errors->first('name') }}</strong>
    </span>
  @endif
</div>
</div>
<div class="form-group row">
  <label for="email" class="col-sm-4 col-form-label text-
md-right">{{ __('Email') }}</label>

  <div class="col-md-6">
    <input id="email" type="email" class="form-control{{
      $errors->has('email') ? ' is-invalid' : '' }}"
    name="email" value="{{ Auth::user()->email }}"
    autofocus>

    @if ($errors->has('email'))
      <span class="invalid-feedback">
        <strong>{{ $errors->first('email') }}</strong>
      </span>
    @endif
  </div>
</div>
<div class="form-group row">
  <label for="avatar" class="col-md-4 col-form-
label text-md-right">{{ __('Avatar') }}</label>

  <div class="col-md-8">
    <input id="avatar" type="file" class="form-
control{{ $errors->has('avatar') ? ' is-invalid'
: '' }}" name="avatar" autofocus>

    @if ($errors->has('avatar'))
      <span class="invalid-feedback">
        <strong>{{ $errors->first('avatar')
        }}</strong>
      </span>
    @endif
  </div>
</div>

<div class="form-group row">
  <label for="password" class="col-md-4 col-form-
label text-md-right">{{ __('Contrasenya')
}}</label>
  <div class="col-md-6">

```

```

        <input id="password" type="password" class="form-
        control{{ $errors->has('password') ? ' is-
        invalid' : '' }}" name="password" required>
        @if ($errors->has('password'))
            <span class="invalid-feedback">
                <strong>{{ $errors->first('password')
                }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group row mb-0">
    <div class="col-md-8 col-md-offset-4">
        <button type="submit" class="btn btn-
        primary">
            {{ __('Guardar els canvis') }}
        </button>
    </div>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
@endsection

```

A.3.20. presentacio/contacte

```
@extends('layouts.plantilla')
```

```
@section('page_title', setting('site.title') . " - " . "Informació de
contacte")
```

```
@section('content')
```

```

<div>
    <h1>Dades de contacte</h1>
</div>
<div>
    <ul class="list-group">
        <li class="list-group-item">
            <h4 class="list-group-item-heading">Adreça postal</h4>
            <p class="list-group-item-text">{{ setting('site.direccio') }}</p>
        </li>
        <li class="list-group-item">
            <h4 class="list-group-item-heading">Horari d'obertura</h4>
            <p class="list-group-item-text">de 10 a 14h de dilluns a
            divendres</p>
        </li>
        <li class="list-group-item">
            <h4 class="list-group-item-heading">Telèfon</h4>
            <p class="list-group-item-text">611223344</p>
        </li>
    </ul>
</div>

```

```

    <p class="list-group-item-text small"> (horari d'atenció: de 10 a
    14h de dilluns a divendres)</p>
  </li>
  <li class="list-group-item">
    <h4 class="list-group-item-heading">Correu electrònic</h4>
    <p class="list-group-item-text">*****@****.com</p>
  </li>
  <li class="list-group-item">
    <h4 class="list-group-item-heading">Xarxes socials</h4>
    <p class="list-group-item-text">(Icones de xarxes socials)
    </p>
  </li>
</ul>
<br>
</div>
<div>
  <h2>o pots posar-te en contacte directament a través del següent
  formulari</h2>
  <p class="small">Tots els camps són necessaris</p>
</div>

<div>
  <form action="https://formcarry.com/s/*****" method="POST" accept-
  charset="UTF-8">

    <div class="row">
      <div class="col-md-6">
        <label for="nomcontacte">Nom</label>
        <input type="text" class="form-control" maxlength="50" name="nom"
        id="nomcontacte" placeholder="Nom amb el que vols que ens dirigim a
        tu" aria-describedby="basic-addon1" required>
      </div>
      <div class="col-md-6">
        <label for="emailcontacte">Adreça electrònica</label>
        <input type="email" class="form-control" maxlength="50"
        name="email" id="emailcontacte" placeholder="La teva adreça de
        correu electrònic" aria-describedby="basic-addon1" required> <!--
        <input type="email" name="email"> use this to reply visitors and
        prevent spam via akismet -->
      </div>
    </div>
    <br>
    <div>
      <label for="missatgecontacte">Missatge</label>
      <textarea class="form-control" rows="10" maxlength="2000"
      name="missatge" id="missatgecontacte" placeholder="Escriu aquí el
      text que ens vulguis enviar" required></textarea>
    </div>
    <div class="checkbox">
      <label>
        <input type="checkbox" name="condicions" id="condicions" required>
        Clicant en aquesta casella acceptes les condicions d'ús. <small>La
        informació enviada serà gestionada parcialment per formcarry. Les
        teves dades de contacte s'utilitzaran únicament per contestar-te i
        per guardar un registre de l'enviament. Aquí pots consultar les <a
        href="{{ route('condicions') }}">{{ __("condicions d'ús") }}</a> de
        {{ setting('site.title') }} més extensament i aquí pots trobar <a

```

```

        href="{{ url('condicions') }}">{{ __("la política de privacitat de
        formcarry") }}</a>. </small><br>
    </label>
</div>
<input type="hidden" name="_gotcha"><!-- use this to prevent spam -->
<input type="submit" class="btn btn-default center-block btn-lg"
value="Envia">
<br>

</form>

<br>
<br>
</div>

@endsection

```

A.3.21. presentacio/equip

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "L'equip darrera
el projecte")

@section('content')

<div>
    <h1>Les persones darrera el projecte</h1>
    <p>En aquest moment som cinc persones treballant amb ganes per tirar
    endavant el projecte de {{ setting('site.title') }}. (Més text de
    presentació)</p><br>
</div>
<div>
    <ul class="list-group">
        <div class="row">
            <li class="list-group-item col-md-6">
                <h4 class="list-group-item-heading">Persona 1</h4>
                <p class="list-group-item-text">Presentació personal</p>
            </li>
            <li class="list-group-item col-md-6">
                <h4 class="list-group-item-heading">Persona 2</h4>
                <p class="list-group-item-text">Presentació personal</p>
            </li>
        </div>
        <div class="row">
            <li class="list-group-item col-md-6">
                <h4 class="list-group-item-heading">Persona 3</h4>
                <p class="list-group-item-text">Presentació personal</p>
            </li>
            <li class="list-group-item col-md-6">
                <h4 class="list-group-item-heading">Persona 4</h4>
                <p class="list-group-item-text">Presentació personal</p>
            </li>
        </div>
        <div class="row">

```



```

        <li class="list-group-item col-md-6">
          <h4 class="list-group-item-heading">Persona 5</h4>
          <p class="list-group-item-text">Presentació personal</p>
        </li>
      </div>
    </ul>
    <br>
  </div>
  <br>
  <br>

@endsection

```

A.3.22. presentacio/onsom

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "On som:
localització de l'espai")

@section('content')

<div class="cos clearfix">
  <div class="container">
    <h1>On som</h1>
    <div>
      <ul class="list-group">
        <li class="list-group-item">
          <h4>Adreça</h4>
          <p>{{ setting('site.direccio') }}</p>
        </li>
        <li class="list-group-item">
          <h4>Com arribar</h4>
          <p>Metro: (metro)</p>
          <p>Tram: (tram)</p>
          <p>Bus: (bus)</p>
          <p>Bus nocturn: (bus nocturn)</p>
        </li>
      </ul>
    <br>
    <h2>Mapa</h2>
    <div class="embed-responsive embed-responsive-16by9">
      <iframe class="embed-responsive-item" src="..." style="border:0"
allowfullscreen></iframe>
    </div>
    <br>
    <br>
  </div>
</div>

@endsection

```

A.3.23. presentacio/projecte

```
@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "El projecte cooperatiu")

@section('content')

<article>
  <h1>Qui som?</h1>
  <p>(Text de presentació)</p>
</article>

<br><br>

@endsection
```

A.3.24. serveis/cotreball

```
@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Servei de cotreball / coworking")

@section('content')

<article>
  <h1>Servei de cotreball</h1>
  <br>
  <p>Oferim un espai de cotreball perquè puguis venir a desenvolupar els teus projectes amb nosaltres.</p>
  <br>

  <ul class="list-group">
    <li class="list-group-item">
      <h4 class="list-group-item-heading">Per a habituals</h4>
      <p class="list-group-item-text">(Horari)</p>
      <p class="list-group-item-text">(Serveis)</p>
      <p class="list-group-item-text"><strong>(Preu)</strong></p>
    </li>
    <li class="list-group-item">
      <h4 class="list-group-item-heading">A mitja jornada</h4>
      <p class="list-group-item-text">(Horari)</p>
      <p class="list-group-item-text">(Serveis)</p>
      <p class="list-group-item-text"><strong>(Preu)</strong></p>
    </li>
    <li class="list-group-item">
      <h4 class="list-group-item-heading">Un dia</h4>
      <p class="list-group-item-text">(Horari)</p>
      <p class="list-group-item-text">(Serveis)</p>
      <p class="list-group-item-text"><strong>(Preu)</strong></p>
    </li>
  </ul>
  <br>
```

```

    <p>Si t'interessa, posat en <a href="{{ route('contacte') }}">{{
    __('contacte amb nosaltres') }}</a> o vine a veure'ns.</p>

</article>
<br><br>

@endsection

```

A.3.25. serveis/espais

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Els nostres
espais - Servei de reserva d'espais")

@section('content')

<article>
    <h1>Els nostres espais</h1>
    <br>
    <p>Tenim diverses sales i espais que oferim per a tallers, xerrades,
    cursos, reunions... Si vols utilitzar-ne alguna, posat en <a href="{{
    route('contacte') }}">{{ __('contacte amb nosaltres') }}</a> o vine a
    veure'ns.</p>
    <br>

    <ul class="list-group">
        @foreach($sales as $sala)
            <li class="list-group-item">
                <h4 class="list-group-item-heading">{{ $sala->nom }}</h4>
                <p class="list-group-item-text">{{ $sala->descripcio }}</p>
                <p class="list-group-item-text">Amb capacitat per a {{
                $sala->capacitat }} persones.</p>
            </li>
        @endforeach
    </ul>

    <br>
</article>
<br><br>

@endsection

```

A.3.26. serveis/exposicions

```

@extends('layouts.plantilla')

@section('page_title', setting('site.title') . " - " . "Exposicions")

@section('content')

```

```

<article>
  <h1>Les exposicions</h1>
  <br>
  <p>A {{ setting('site.title') }} creiem en la importància de l'art a la
vida de les persones. Per això volem donar suport a artistes facilitant
una zona d'exposició per a les seves obres. Si és el teu cas i vols
exposar, posat en <a href="{{ route('contacte') }}">{{ __( 'contacte amb
nosaltres' ) }}</a> o vine a veure'ns.</p>
  <br>
  <p>Actualment no tenim cap exposició en mostra, però esperem tenir-ne
aviat.</p>
</article>
<br><br>

@endsection

```

A.3.27. layouts/plantilla

```

<!DOCTYPE html>
<html lang="ca"> <!-- català -->

<head>

  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

  <title>@yield('page_title', setting('site.title') . " - " .
setting('site.description'))</title>

  <!-- Favicon -->
  <link rel="shortcut icon" href="{{ url('/images/IconLogo.png') }}"
type="image/x-icon">

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.c
ss" integrity="sha384-
BVYiISIFeKldGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
  <link href="{{ asset('css/estil.css') }}" rel="stylesheet">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"><
/script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIPG9mGCD8wGNICPD7Txa"
crossorigin="anonymous"></script>
</head>

<body>

<div class="caixa">
<header class="clearfix">

```

```

<div class="container">
  @yield("header")
  
</div>
</header>

@include("layouts.navbar")

<div class="cos clearfix">
  <div class="container">
    @yield("content")
  </div>
</div>

<footer>
  <div class="container">
    <div class="row row-footer row-dark">
      <p>Consulta les <a href="{{ route('condicions') }}">condicions
        d'ús</a> d'aquest lloc web</p>
    </div>
    <div class="row row-footer row-light">
      <p><a href="{{ route('contacte') }}">Contacte amb
        nosaltres.</a></p>
    </div>
  </div>
</footer>
</div>
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
  integrity="sha384-
  KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
  crossorigin="anonymous"></script>
<script
  src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.m
  in.js" integrity="sha384-
  ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
  crossorigin="anonymous"></script>
<script
  src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
  integrity="sha384-
  JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
  crossorigin="anonymous"></script>

</body>
</html>

```

A.3.28. layouts/navbar

```

<nav class="navbar navbar-default"> <!-- barra navegacio de bootstrap -->
  <div class="container">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-
        toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
        expanded="false">
        <span class="sr-only">Desplegable de navegació</span>
    </div>
  </div>

```

```

        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="{{ url('/') }}" title="Inici"><span
    class="sr-only"></span><i class="glyphicon glyphicon-home"></i></a>
</div>

<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse"
id="bs-example-navbar-collapse-1">
    <ul class="nav navbar-nav">
        <li class="dropdown" title="Presentació">
            <a class="dropdown-toggle" data-toggle="dropdown" role="button"
            aria-haspopup="true" aria-expanded="false">Presentació <span
            class="caret"></span></a>
            <ul class="dropdown-menu">
                <li><a href="{{ url('/projecte') }}">El projecte</a></li>
                <li><a href="{{ url('/equip') }}">L'equip</a></li>
                <li role="separator" class="divider"></li>
                <li><a href="{{ url('/onsom') }}" title="Mapa">On
                som</a></li>
                <li role="separator" class="divider"></li>
                <li><a href="{{ url('/contacte') }}">Contacte</a></li>
            </ul>
        </li>
        <li class="dropdown" title="Serveis">
            <a class="dropdown-toggle" data-toggle="dropdown" role="button"
            aria-haspopup="true" aria-expanded="false">Serveis <span
            class="caret"></span></a>
            <ul class="dropdown-menu">
                <li><a href="{{ url('/espais') }}">Reservar espais</a></li>
                <li><a href="{{ url('/cotreball') }}">Cotreball</a></li>
                <li><a href="{{ url('/exposicions') }}">Exposicions</a></li>
            </ul>
        </li>
        <li class="dropdown">
            <a href="activitats.html" class="dropdown-toggle" data-
            toggle="dropdown" role="button" aria-haspopup="true" aria-
            expanded="false">Activitats <span class="caret"></span></a>
            <ul class="dropdown-menu">
                <li><a href="{{ route('activitats') }}">Activitats
                programades</a></li>
                <li role="separator" class="divider"></li>
                <li><a href="{{ route('crear-activitat') }}">Proposa una
                activitat</a></li>
            </ul>
        </li>
        <li>
            <a href="{{ url('/articles') }}">Notícies</a>
        </li>
        <li>
            <a href="{{ route('directori') }}">Fem xarxa!</a>
        </li>
    </ul>

    <ul class="nav navbar-nav navbar-right">
        <li class="dropdown">

```

```

    @if (Auth::guest())
    <a class="dropdown-toggle" title="Inicia sessió o registra't"
    data-toggle="dropdown" role="button" aria-haspopup="true" aria-
    expanded="false"><i class="glyphicon glyphicon-user"></i></a>
    <ul class="dropdown-menu">

    <!-- Authentication Links -->
    <li><a href="{{ route('login') }}">Iniciar sessió</a></li>
    <li><a href="{{ route('register') }}">Registrar-se</a></li>
    </ul>

    @else
    <a class="dropdown-toggle" title="{{ Auth::user()->name }}"
    data-toggle="dropdown" role="button" aria-haspopup="true" aria-
    expanded="false"><i class="glyphicon glyphicon-user"></i></a>
    <ul class="dropdown-menu">

    <!-- ADD ROLES AND USERS LINKS -->
    @if (Voyager::can('browse_admin'))
    <li><a href="{{ route('voyager.login') }}">Admin</a></li>
    <li role="separator" class="divider"></li>
    @endif

    <li><a href="{{ route('perfil') }}"
    onclick="event.preventDefault();
    document.getElementById('perfil-form').submit();">El
    meu perfil</a>

    <form id="perfil-form" action="{{ route('perfil') }}"
    method="POST" style="display: none;">
    {{ csrf_field() }}
    </form>
    </li>
    <li><a href="{{ route('perfil-activitats') }}"
    onclick="event.preventDefault();
    document.getElementById('perfil-act-
    form').submit();">Les meves activitats</a>

    <form id="perfil-act-form" action="{{ route('perfil-
    activitats') }}" method="POST" style="display: none;">
    {{ csrf_field() }}
    </form>
    </li>
    @if (Auth::user()->directori !== null)
    <li><a href="{{ route('perfil-directori') }}"
    onclick="event.preventDefault();
    document.getElementById('perfil-dir-
    form').submit();">El meu directori</a>

    <form id="perfil-dir-form" action="{{ route('perfil-
    directori') }}" method="POST" style="display: none;">
    {{ csrf_field() }}
    </form>
    </li>
    @endif
    <li role="separator" class="divider"></li>
    <li>

```

```

        <a href="{{ route('logout') }}"
            onclick="event.preventDefault();
            document.getElementById('logout-form').submit();"
            Logout
        </a>

        <form id="logout-form" action="{{ route('logout') }}"
            method="POST" style="display: none;"
            {{ csrf_field() }}
        </form>
    </li>
</ul>
@endif
</li>

</ul><!-- /.navbar-right -->
</div><!-- /.navbar-collapse -->
</div><!-- /.container -->
</nav>

```

A.3.29. errors/403

```

<?php ?>

@extends('layouts.plantilla')

@section('content')
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <h1>No tens permisos per veure aquesta pàgina</h1>
        </div>
    </div>
@endsection

```

A.3.30. errors/404

```

<?php ?>

@extends('layouts.plantilla')

@section('content')
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <h1>Ups! No trobem aquesta pàgina :S</h1>
        </div>
    </div>
@endsection

```


A.4. Controladors

A.4.1. ActivitatController

```
<?php
```

```
namespace App\Http\Controllers;

use App\Activitat;
use App\User;
use App\Sales;
use Illuminate\Support\Facades\DB;
use App\Http\Controllers\Controller;
use Carbon\Carbon;
use Illuminate\Http\Request;
use TCG\Voyager\Models\DataType;
use Illuminate\Http\Response;

class ActivitatController extends Controller
{
    public function show($slug)
    {
        $activitat = Activitat::where('slug', '=', $slug)
            ->where('estat', '=', 'Publicada')
            ->whereNull('deleted_at')
            ->firstOrFail();

        $assistents = $activitat->assistents()->get();
        $responsables = $activitat->responsables()->get();

        if ($activitat->sala == null) {
            $sala = "Espai pendent de decidir";
        } else {
            $sala = Sales::find($activitat->sala)->nom;
        }

        if ($activitat->data_final == null) {
            return view('activitats.activundia', compact('activitat'))
                ->withSala($sala)->withAssistents($assistents)
                ->withResponsables($responsables);
        } else {
            if ($activitat->data_final == $activitat->data_inici) {
                return view('activitats.activundia',
                    compact('activitat'))->withSala($sala)
                    ->withAssistents($assistents)
                    ->withResponsables($responsables);
            } else if ($activitat->tipus_act == 'Jornades') {
                return view('activitats.jornades',
                    compact('activitat'))->withSala($sala)
                    ->withAssistents($assistents)
                    ->withResponsables($responsables);
            } else {
                return view('activitats.curs',
                    compact('activitat'))->withSala($sala)
                    ->withAssistents($assistents)
                    ->withResponsables($responsables);
            }
        }
    }
}
```

```

    }

    public function index()
    {
        $dt = Carbon::today();
        $activitats = Activitat::whereDate('data_inici', '>=', $dt)
            ->where('estat', '=', 'Publicada')
            ->whereNull('deleted_at')
            ->orderBy('data_inici', 'asc')
            ->orderBy('hora_inici', 'asc')
            ->get();

        if ($activitats->isEmpty()) {
            return view('activitats.senseactivitats');
        } else {
            return view('activitats.allactivitats',
                compact('activitats'));
        }
    }
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    return view('/activitats/crear');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request, $id_user)
{
    $this->validate($request, [
        'titol' => 'required|max:255',
        'data_inici' => 'nullable|date',
        'data_final' => 'nullable|date',
        'horari' => 'nullable|max:300',
        'places_max' => 'nullable|max:4|number',
        'resum' => 'nullable|max:1000',
        'descr_llarga' => 'nullable|max:5000',
        'imatge' =>
            'nullable|image|dimensions:min_width=100,min_height=100|max:400',
        //max: en kB
        'condicions' => 'required'
    ]);

    $activitat = new Activitat;

    $activitat->responsables()->attach($id_user);
}

```

```

    $activitat->titol = $request->titol;

    if ($request->hasFile('imatge')){

        $imatge = $request->file('imatge');
        $filename = $imatge->getClientOriginalName();
        $path = $imatge->storeAs('public/activitats', $filename);

        $activitat->imatge = 'activitats/' . $filename;
    }

    $activitat->tipus_act = $request->tipus_act;
    $activitat->data_inici = $request->data_inici;
    $activitat->data_final = $request->data_final;
    $activitat->horari = $request->horari;
    $activitat->places_max = $request->places_max;
    $activitat->taquilla_inv = $request->taquilla_inv;
    $activitat->resum = $request->resum;
    $activitat->descr_llarga = $request->descr_llarga;
    $activitat->save();

    $missatge = "Gràcies per enviar la teva proposta, en breu la
    revisarem :)";

    return view('activitat.editor', compact('activitat'))
    ->withMissatge($missatge);
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit(Activitat $activitat)
{
    if ($activitat->deleted_at == null) {
        $missatge = null;
        return view('activitats.editor', compact('activitat'))
        ->withMissatge($missatge);
    }
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id

```

```
* @return \Illuminate\Http\Response
*/
public function update(Request $request, $id)
{
    $this->validate($request, [
        'titol' => 'required|max:255',
        'data_inici' => 'nullable|date',
        'data_final' => 'nullable|date',
        'horari' => 'nullable|max:300',
        'places_max' => 'nullable|max:4|number',
        'resum' => 'nullable|max:1000',
        'descr_llarga' => 'nullable|max:5000',
        'imatge' =>
'nullable|image|dimensions:min_width=100,min_height=100|max:400',
//max: en kB
        'condicions' => 'required'
    ]);

    $activitat = Activitat::find($id);

    $this->authorize('update', $activitat);

    $activitat->titol = $request->titol;
    if ($activitat->data_inici !== null){
        $activitat->data_inici = $request->data_inici;
    }
    if ($activitat->data_final !== null){
        $activitat->data_final = $request->data_final;
    }
    if ($activitat->horari !== null){
        $activitat->horari = $request->horari;
    }

    $activitat->tipus_act = $request->tipus_act;

    $activitat->places_max = $request->places_max;

    $activitat->taquilla_inv = $request->taquilla_inv;

    $activitat->resum = $request->resum;

    $activitat->descr_llarga = $request->descr_llarga;

    if ($request->hasFile('imatge')){

        Storage::delete("public/" . $activitat->imatge);
        $imatge = $request->file('imatge');
        $filename = $imatge->getClientOriginalName();
        $path = $imatge->storeAs('public/activitats', $filename);

        $activitat->imatge = 'activitats/' . $filename;
    }

    $activitat->save();

    $missatge = "Canvis guardats!";
}
```

```

        return view('activitat.editor', compact('activitat'))
        ->withMissatge($missatge);
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy(Request $request, Activitat $activitat)
    {
        $this->authorize('update', $activitat);
        $activitat->delete();
        return redirect()->route('activitats');
    }

    //vincula l'activitat amb el user-assistent
    public function atAssistent($id_act, $id_user)
    {
        $activitat = Activitat::find($id_act);
        $activitat->assistents()->attach($id_user);
        return redirect()->route('showActivitat', $activitat->slug);
    }

    //desvincula l'activitat del user assistent
    public function detAssistent($id_act, $id_user)
    {
        $activitat = Activitat::find($id_act);
        $activitat->assistents()->detach($id_user);
        return redirect()->route('showActivitat', $activitat->slug);
    }
}

```

A.4.2. DirectoriController

```

<?php

namespace App\Http\Controllers;

use App\Directorio;
use App\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;
use Illuminate\Http\File;

class DirectoriController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $entrades = Directorio::whereNull('deleted_at')

```

```
->where('estat', '=', 'actiu')
->orderBy('tipus', 'asc')
->orderBy('nom', 'asc')
->get();
if ($entrades->isEmpty()) {
    return view('directori.senseentrades');
} else {
    return view('directori.index', compact('entrades'));
}
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    return view('directori.nova-entrada');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request, $id_user)
{
    $directori = new Directori;

    $this->validate($request, [
        'nom' => 'required|max:255',
        'web' => 'nullable|string|max:255',
        'email' => 'nullable|string|email|max:255',
        'adreca' => 'nullable|max:400',
        'descripcio' => 'nullable|max:1000',
        'imatge' =>
            'nullable|image|dimensions:min_width=100,min_height=100|max:200',
        //max: en kB
        'condicions' => 'required'
    ]);

    $directori->user_id = $id_user;

    $directori->nom = $request->nom;

    $directori->web = $request->web;

    $directori->email = $request->email;

    $directori->adreca = $request->adreca;

    $directori->descripcio = $request->descripcio;
```

```

        if ($request->hasFile('imatge')){

            $imatge = $request->file('imatge');
            $filename = $imatge->getClientOriginalName();
            $path = $imatge->storeAs('public/directori', $filename);
            $directori->imatge = 'directori/' . $filename;
        }

        $directori->save();

        $missatge = "Gràcies per enviar la teva proposta, en breu la
        revisarem :)";

        return view('directori.editar', compact('directori'))
        ->withMissatge($missatge);
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit(Directori $directori)
    {
        if ($directori->deleted_at == null) {
            $missatge = null;
            return view('directori.editar', compact('directori'))
            ->withMissatge($missatge);
        }
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, Directori $directori)
    {
        $this->authorize('update', $directori);
        $this->validate($request, [
            'nom' => 'required|max:255',
            'web' => 'nullable|string|max:255',
            'email' => 'nullable|string|email|max:255',
            'adreca' => 'nullable|max:400',
            'descripcio' => 'nullable|max:1000',
            'imatge' =>
            'nullable|image|dimensions:min_width=100,min_height=100|max:200',
            //max: en kB
            'condicions' => 'required'
        ]);

        $directori->nom = $request->nom;

        $directori->web = $request->web;
    }

```

```

        $directori->email = $request->email;

        $directori->adreca = $request->adreca;

        $directori->descripcio = $request->descripcio;

        if ($request->hasFile('imatge')){

            Storage::delete("public/" . $directori->imatge);
            $imatge = $request->file('imatge');
            $filename = $imatge->getClientOriginalName();
            $path = $imatge->storeAs('public/directori', $filename);

            $directori->imatge = 'directori/' . $filename;

        }

        $directori->save();
        $missatge = "Canvis guardats!";

        return view('directori.editar', compact('directori'))
            ->withMissatge($missatge);
    }
    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy(Request $request, Directori $directori)
    {
        $this->authorize('update', $directori);
        $directori->delete();
        return redirect()->route('directori');
    }
}

```

A.4.3. IndexController

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Activitat;
use TCG\Voyager\Models\Post;
use Carbon\Carbon;

class IndexController extends Controller
{
    public function __invoke()
    {
        $dt = Carbon::today();
        $activitats = Activitat::whereDate('data_inici', '>=', $dt)
    }
}

```



```

        ->where('estat', '=', 'Publicada')
        ->whereNull('deleted_at')
        ->orderBy('data_inici', 'asc')
        ->orderBy('hora_inici', 'asc')
        ->take(5)
        ->get();
    $posts = Post::where('status', '=', 'PUBLISHED')
        ->whereNull('deleted_at')
        ->orderBy('created_at', 'desc')
        ->take(5)
        ->get();
    $destacat = Activitat::where('estat', '=', 'Publicada')
        ->whereDate('data_inici', '>=', $dt)
        ->whereNull('deleted_at')
        ->where('destacada', '=', 'SI')
        ->orderBy('data_inici', 'asc')
        ->first();
    if($destacat == null) {
        $destacat = Post::where('status', '=', 'PUBLISHED')
            ->whereNull('deleted_at')
            ->where('featured', '=', '1')
            ->orderBy('created_at', 'desc')
            ->first();
    }
    return view('index', compact('activitats'), compact('posts'))
    ->withDestacat($destacat);
}
}

```

A.4.4. PerfilController

```
<?php
```

```

namespace App\Http\Controllers;

use App\User;
use App\Activitat;
use App\Directorio;
use TCG\Voyager\Models\Role;
use Illuminate\Http\Request;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Support\Facades\Auth;
use Carbon\Carbon;

class PerfilController extends Controller
{
    public function perfil(Request $request)
    {
        $user = $request->user();
        $dt = Carbon::today();
        $activresp = $user->responsables()
            ->whereNull('deleted_at')
            ->whereDate('data_inici', '>=', $dt)
            ->get(); //retorna activitats
        $activassis = $user->assistents()
    }
}

```

```

        ->whereNull('deleted_at')
        ->where('estat', '=', 'Publicada')
        ->whereDate('data_inici', '>=', $dt)
        ->get(); //retorna activitats

    $directori = $user->directori;

    return view('perfil.perfil', compact('user'))
    ->withActivassis($activassis)->withActivresp($activresp)
    ->withDirectori($directori);
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit()
{
    return view('perfil.editar');
}

use AuthenticatesUsers;

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request)
{
    $this->validate($request, [
        'name' => 'required|string|max:255',
        'email' => 'required|string|email|max:255',
        'password' => 'required|string|min:6'
    ]); //validator($request->all())->validate();

    if ($this->hasTooManyLoginAttempts($request)) {
        $this->fireLockoutEvent($request);
        return $this->sendLockoutResponse($request);
    }

    $user = $request->user();

    if (Auth::attempt(['email' => $user->email, 'password' =>
    $request->password])) {
        // The user is active, and exists.
        $this->clearLoginAttempts($request);
        $user->name = $request->name;
        $user->email = $request->email;
        $user->avatar = $request->avatar;
        $user->save();

        return redirect()->route('home');
    }
}

```

```

        $this->incrementLoginAttempts($request);

        return $this->sendFailedLoginResponse($request);
    }

    public function esborra() //retorna la pàgina de confirmació per a
l'esborrat
    {
        return view('perfil.esborrar');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy(Request $request)
    {
        $this->validate($request, ['password' =>
        'required|string|min:6']);

        if ($this->hasTooManyLoginAttempts($request)) {
            $this->fireLockoutEvent($request);

            return $this->sendLockoutResponse($request);
        }

        $user = $request->user();

        if (Auth::attempt(['email' => $user->email, 'password' =>
        $request->password])) {
            // The user is active, and exists.
            $this->clearLoginAttempts($request);

            // Detach all from the user...
            $user->assistents()->detach();
            $user->responsables()->detach();
            $user->directori()->dissociate();

            if (($user->avatar !== 'users/default.png') &&
            ($user->avatar !== null)){
                Storage::delete("public/" . $user->avatar);
            }

            $user->delete();

            return redirect()->route('index');
        }

        $this->incrementLoginAttempts($request);

        return $this->sendFailedLoginResponse($request);
    }
}

```

A.4.5. PostController

```
<?php

namespace App\Http\Controllers;

use TCG\Voyager\Models\DataType;
use TCG\Voyager\Models\Post;
use App\User;
use Illuminate\Http\Request;

class PostController extends Controller
{
    public function show($slug) {
        $post = Post::where('slug', '=', $slug)
            ->where('status', '=', 'PUBLISHED')
            ->whereNull('deleted_at')
            ->firstOrFail();
        if ($post->author_id == null) {
            $author = "";
        } else {
            $author = $post->authorId->name;
        }

        return view('articles.post', compact('post'))
            ->withAuthor($author);
    }

    public function index() {
        $posts = Post::where('status', '=', 'PUBLISHED')
            ->whereNull('deleted_at')
            ->orderBy('created_at', 'desc')
            ->take(10)
            ->get();
        return view('articles.allposts', compact('posts'));
    }
}
```

A.5. Polítiques d'autorització

A.5.1. ActivitatPolicy

```
<?php

namespace App\Policies;

use App\User;
use App\Activitat;
use Illuminate\Auth\Access\HandlesAuthorization;
use TCG\Voyager\Policies\BasePolicy;

class ActivitatPolicy extends BasePolicy
{
    use HandlesAuthorization;

    /**
```

```

* Determine whether the user can view the activitat.
*
* @param \App\User $user
* @param \App\Activitat $activitat
* @return mixed
*/
public function view(User $user, Activitat $activitat)
{
    $responsables = $user->responsables()->get();
    $comprova = true;

    foreach ($responsables as $responsable) {
        if ($responsable->id == $activitat->id){
            $comprova = false;
            return true;
        }
    }

    if ($comprova = true) {
        return false;
    }
}
/**
* Determine whether the user can update the activitat.
*
* @param \App\User $user
* @param \App\Activitat $activitat
* @return mixed
*/
public function update(User $user, Activitat $activitat)
{
    $responsables = $activitat->responsables()->get();
    $comprova = true;

    foreach ($responsables as $responsable) {
        if ($responsable->id == $user->id){
            $comprova = false;
            return true;
        }
    }

    if ($comprova = true) {
        return false;
    }
}
/**
* Determine whether the user can delete the activitat.
*
* @param \App\User $user
* @param \App\Activitat $activitat
* @return mixed
*/
public function delete(User $user, Activitat $activitat)
{
    if ($user->role->name == ('admin' or 'editor')) {
        return true;
    }
}

```

```

        $responsables = $activitat->responsables()->get();
        $comprova = true;

        foreach ($responsables as $responsable) {
            if ($responsable->id == $user->id){
                $comprova = false;
                return true;
            }
        }

        if ($comprova = true) {
            return false;
        }
    }
}

```

A.5.2. DirectoriPolicy

```

<?php

namespace App\Policies;

use App\User;
use App\Director;
use Illuminate\Auth\Access\HandlesAuthorization;
use TCG\Voyager\Policies\BasePolicy;

class DirectorPolicy extends BasePolicy
{
    use HandlesAuthorization;

    /**
     * Determine whether the user can create directoris.
     *
     * @param \App\User $user
     * @return mixed
     */
    public function create(User $user)
    {
        if ($user->director) {
            return false;
        }
        return true;
    }

    /**
     * Determine whether the user can update the directori.
     *
     * @param \App\User $user
     * @param \App\Director $director
     * @return mixed
     */
    public function update(User $user, Director $director)
    {

```

```

        return $user->id === $directori->user_id;
    }

    /**
     * Determine whether the user can delete the directori.
     *
     * @param \App\User $user
     * @param \App\Directori $directori
     * @return mixed
     */
    /**public function delete(User $user, Directori $directori)
    {
        return $user->id === $directori->user_id;
    }*/
}

```

A.6. Notificador

A.6.1. RresetPasswordCat

```
<?php
```

```

namespace App\Notifications;

use Illuminate\Bus\Queueable;
use Illuminate\Notifications\Notification;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Notifications\Messages\MailMessage;

class ResetPasswordCat extends Notification
{
    /**
     * The password reset token.
     *
     * @var string
     */
    public $token;

    /**
     * The callback that should be used to build the mail message.
     *
     * @var \Closure|null
     */
    public static $toMailCallback;

    /**
     * Create a notification instance.
     *
     * @param string $token
     * @return void
     */
    public function __construct($token)
    {

```

```

        $this->token = $token;
    }

    /**
     * Get the notification's channels.
     *
     * @param mixed $notifiable
     * @return array|string
     */
    public function via($notifiable)
    {
        return ['mail'];
    }

    /**
     * Build the mail representation of the notification.
     *
     * @param mixed $notifiable
     * @return \Illuminate\Notifications\Messages\MailMessage
     */
    public function toMail($notifiable)
    {
        if (static::$toMailCallback) {
            return call_user_func(static::$toMailCallback, $notifiable,
                $this->token);
        }

        return (new MailMessage)
            ->subject("Reestablir contrasenya")
            ->greeting("Hola " . $notifiable->name)
            ->line("Estàs rebent aquest email perquè hem rebut una sol·licitud per reestablir la contrasenya del teu compte.")
            ->action('Reestablir contrasenya',
                url(config('app.url').route('password.reset', $this->token, false)))
            ->line("Si tu no has fet aquesta sol·licitud, no cal que facis res.")
            ->salutation("Cordialment, La Trobada");
    }

    /**
     * Set a callback that should be used when building the notification mail message.
     *
     * @param \Closure $callback
     * @return void
     */
    public static function toMailUsing($callback)
    {
        static::$toMailCallback = $callback;
    }
}

```


A.6.2. email.blade

```

@component('mail::message')
{{-- Greeting --}}
@if (! empty($greeting))
# {{ $greeting }}
@else
@if ($level == 'error')
# Whoops!
@else
# Hello!
@endif
@endif

{{-- Intro Lines --}}
@foreach ($introLines as $line)
{{ $line }}
@endforeach

{{-- Action Button --}}
@isset($actionText)
<?php
    switch ($level) {
        case 'success':
            $color = 'green';
            break;
        case 'error':
            $color = 'red';
            break;
        default:
            $color = 'blue';
    }
?>
@component('mail::button', ['url' => $actionUrl, 'color' => $color])
{{ $actionText }}
@endcomponent
@endisset

{{-- Outro Lines --}}
@foreach ($outroLines as $line)
{{ $line }}
@endforeach

{{-- Salutation --}}
@if (! empty($salutation))
{{ $salutation }}
@else
Salutacions,<br>{{ config('app.name') }}
@endif

{{-- Subcopy --}}
@isset($actionText)
@component('mail::subcopy')
Si tens problemes obrint l'enllaç "{{ $actionText }}", copia i enganxa
això al teu navegador: [{{ $actionUrl }}]({{ $actionUrl }})
@endcomponent
@endisset
@endcomponent

```